What is a tensor? (Part II)

Lek-Heng Lim

University of Chicago

goals of these tutorials

- very elementary introduction to tensors
- through the lens of linear algebra and numerical linear algebra
- highlight their roles in computations
- trace chronological development through three definitions
  ① a multi-indexed object that satisfies tensor transformation rules
  ② a multilinear map
  ③ an element of a tensor product of vector spaces
- last week: ① — trickiest one to discuss
- today: ② and ③ — much easier modern definitions

recap from part I

- take home idea: **a tensor is defined by its transformation rule**
- example: is this

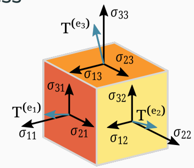
$$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 3 & 4 & 5 \end{bmatrix}$$

  a tensor?
- makes no sense
- definition ① requires a context

recap from part I

- if it comes from a measurement of stress

$$\begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix}$$



  then it is a contravariant 2-tensor

- if we are interested in its eigenvalues and eigenvectors

$$(XAX^{-1})Xv = \lambda Xv$$

  then it is a mixed 2-tensor

- if we are interested in its Hadamard product

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \circ \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} = \begin{bmatrix} a_{11}b_{11} & a_{12}b_{12} & a_{13}b_{13} \\ a_{21}b_{21} & a_{22}b_{22} & a_{23}b_{23} \\ a_{31}b_{31} & a_{32}b_{32} & a_{33}b_{33} \end{bmatrix}$$

  then it is not a tensor

recap from part I

- indices tell us nothing
- $A \in \mathbb{R}^{m\times n}$ has two indices but if transformation rule is

$$A' = XA = [Xa_1, \dots, Xa_n] \quad \text{or} \quad A' = X^{-\mathsf{T}}A = [X^{-\mathsf{T}}a_1, \dots, X^{-\mathsf{T}}a_n]$$

  then it is covariant or contravariant 1-tensor respectively
- e.g., Householder QR algorithm, equivariant neural networks — all covariant 1-tensors no matter how many indices
- want coordinate-free definition that does not depend on indices

tensors via multilinear maps

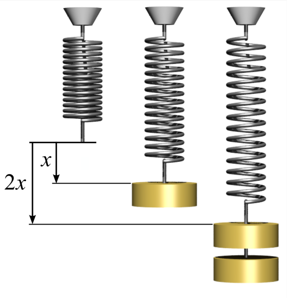
multilinear maps

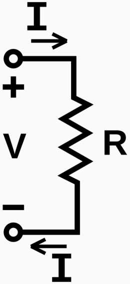
- $\mathbb{V}_1, \dots, \mathbb{V}_d$ and $\mathbb{W}$ vector spaces
- multilinear map or d-linear map is $\varphi : \mathbb{V}_1 \times \cdots \times \mathbb{V}_d \to \mathbb{W}$ with

$$\begin{aligned}\Phi(v, \dots, \lambda v_k + \lambda' v_k', \dots, v_d) \\ = \lambda \Phi(v_1, \dots, v_k, \dots, v_d) + \lambda' \Phi(v_1, \dots, v_k', \dots, v_d)\end{aligned}$$

  for $v_1 \in \mathbb{V}_1, \dots, v_k, v_k' \in \mathbb{V}_k, \dots, v_d \in \mathbb{V}_d$, $\lambda, \lambda' \in \mathbb{R}$
- write $\mathsf{M}^d(\mathbb{V}_1, \dots, \mathbb{V}_d; \mathbb{W})$ for set of all such maps
- write $\mathsf{M}^1(\mathbb{V}; \mathbb{W}) = \mathsf{L}(\mathbb{V}; \mathbb{W})$ for linear maps

why important





linearity principle: almost any natural process is linear in small amounts almost everywhere

multilinearity principle: if we keep all but one factors constant, the varying factor obeys principle of linearity

Hooke's and Ohm's laws both linear but not if we pass a current through the spring or stretch the resistor

$d = 1$

- $\mathscr{B} = \{v_1, \dots, v_m\}$ basis of $\mathbb{V}$, $\mathscr{B}^* = \{v_1^*, \dots, v_m^*\}$ dual basis
- any $v \in \mathbb{V}$ uniquely represented by $a \in \mathbb{R}^m$

$$\mathbb{V} \ni v = a_1 v_1 + \cdots + a_m v_m \quad \longleftrightarrow \quad [v]_{\mathscr{B}} := \begin{bmatrix} a_1 \\ \vdots \\ a_m \end{bmatrix} \in \mathbb{R}^m$$

- any $\varphi \in \mathbb{V}^*$ uniquely represented by $b \in \mathbb{R}^m$

$$\mathbb{V}^* \ni \varphi = b_1 v_1^* + \cdots + b_m v_m^* \quad \longleftrightarrow \quad [\varphi]_{\mathscr{B}^*} := \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix} \in \mathbb{R}^m$$

$d = 1$

- if $\mathscr{C} = \{v_1', \dots, v_m'\}$ another basis and $X \in \mathbb{R}^{m \times m}$ is

$$v_j' = \sum_{i=1}^{m} x_{ij} v_i$$

- change-of-basis theorem:

$$[v]_{\mathscr{C}} = X^{-1}[v]_{\mathscr{B}}, \qquad [\varphi]_{\mathscr{C}^*} = X^{\mathsf{T}}[\varphi]_{\mathscr{B}^*}$$

- recover transformation rules for contravariant and covariant 1-tensors

$$a' = X^{-1}a, \qquad b' = X^{\mathsf{T}}b$$

- vectors $\longleftrightarrow$ contravariant 1-tensors
- linear functionals $\longleftrightarrow$ covariant 1-tensors

$d = 2$

- bases $\mathscr{A} = \{u_1, \ldots, u_n\}$ on $\mathbb{U}$, $\mathscr{B} = \{v_1, \ldots, v_m\}$ on $\mathbb{V}$
- linear operator $\Phi : \mathbb{U} \to \mathbb{V}$ has matrix representation

$$[\Phi]_{\mathscr{A},\mathscr{B}} = A \in \mathbb{R}^{m \times n}$$

  where

$$\Phi(u_j) = \sum_{i=1}^{m} a_{ij} v_i$$

- new bases $\mathscr{A}'$ and $\mathscr{B}'$

$$[\Phi]_{\mathscr{A}',\mathscr{B}'} = A' \in \mathbb{R}^{m \times n}$$

- change-of-basis theorem: if $X \in \mathrm{GL}(m)$ change-of-basis matrix on $\mathbb{V}$, $Y \in \mathrm{GL}(n)$ change-of-basis matrix on $\mathbb{U}$, then

$$A' = X^{-1} A Y$$

$d = 2$

- special case $\mathbb{U} = \mathbb{V}$ with $\mathscr{A} = \mathscr{B}$ and $\mathscr{A}' = \mathscr{B}'$

$$A' = X^{-1} A X$$

- recover transformation rules for mixed 2-tensors
- bilinear functional $\beta : \mathbb{U} \times \mathbb{V} \to \mathbb{R}$ with

$$\beta(\lambda u + \lambda' u', v) = \lambda \beta(u, v) + \lambda' \beta(u', v),$$
$$\beta(u, \lambda v + \lambda' v') = \lambda \beta(u, v) + \lambda' \beta(u, v')$$

  for all $u, u' \in \mathbb{U}$, $v, v' \in \mathbb{V}$, $\lambda, \lambda' \in \mathbb{R}$
- matrix representation of β

$$[\beta]_{\mathscr{A}, \mathscr{B}} = A \in \mathbb{R}^{m \times n}$$

  given by

$$a_{ij} = \beta(u_i, v_j)$$

$d = 2$

- change-of-basis theorem: if

$$[\beta]_{\mathscr{A}',\mathscr{B}'} = A' \in \mathbb{R}^{m \times n},$$

  then

$$A' = X^{\mathsf{T}} A Y$$

- special case $\mathbb{U} = \mathbb{V}$, $\mathscr{A} = \mathscr{B}$, $\mathscr{A}' = \mathscr{B}'$

$$A' = X^{\mathsf{T}} A X$$

- recover transformation rules for covariant 2-tensors
- linear operators $\longleftrightarrow$ mixed 2-tensors
- bilinear functionals $\longleftrightarrow$ covariant 2-tensors

1- and 2-tensor transformation rules

contravariant 1-tensor:	$a' = X^{-1}a$	$a' = Xa$
covariant 1-tensor:	$a' = X^{\mathsf{T}}a$	$a' = X^{-\mathsf{T}}a$
covariant 2-tensor:	$A' = X^{\mathsf{T}}AX$	$A' = X^{-\mathsf{T}}AX^{-1}$
contravariant 2-tensor:	$A' = X^{-1}AX^{-\mathsf{T}}$	$A' = XAX^{\mathsf{T}}$
mixed 2-tensor:	$A' = X^{-1}AX$	$A' = XAX^{-1}$
contravariant 2-tensor:	$A' = X^{-1}AY^{-\mathsf{T}}$	$A' = XAY^{\mathsf{T}}$
covariant 2-tensor:	$A' = X^{\mathsf{T}}AY$	$A' = X^{-\mathsf{T}}AY^{-1}$
mixed 2-tensor:	$A' = X^{-1}AY$	$A' = XAY^{-1}$

$d = 3$

- bilinear operator $\mathrm{B} : \mathbb{U} \times \mathbb{V} \to \mathbb{W}$,

$$\mathrm{B}(\lambda u + \lambda' u', v) = \lambda \mathrm{B}(u, v) + \lambda' \mathrm{B}(u', v),$$
$$\mathrm{B}(u, \lambda v + \lambda' v') = \lambda \mathrm{B}(u, v) + \lambda' \mathrm{B}(u, v')$$

- bases $\mathscr{A} = \{u_1, \dots, u_n\}$, $\mathscr{B} = \{v_1, \dots, v_m\}$, $\mathscr{C} = \{w_1, \dots, w_p\}$,

$$\mathrm{B}(u_i, v_j) = \sum_{i=1}^{p} a_{ijk} w_k$$

- change-of-basis theorem: if

$$[\mathrm{B}]_{\mathscr{A},\mathscr{B},\mathscr{C}} = A \quad \text{and} \quad [\mathrm{B}]_{\mathscr{A}',\mathscr{B}',\mathscr{C}'} = A' \in \mathbb{R}^{m \times n \times p},$$

then

$$A' = (X^{\mathsf{T}}, Y^{\mathsf{T}}, Z^{-1}) \cdot A$$

$d = 3$

- trilinear functional $\tau : \mathbb{U} \times \mathbb{V} \times \mathbb{W} \to \mathbb{R}$,

$$\tau(\lambda u + \lambda' u', v, w) = \lambda\tau(u, v, w) + \lambda'\tau(u', v, w),$$
$$\tau(u, \lambda v + \lambda' v', w) = \lambda\tau(u, v, w) + \lambda'\tau(u, v', w),$$
$$\tau(u, v, \lambda w + \lambda' w') = \lambda\tau(u, v, w) + \lambda'\tau(u, v, w')$$

- bases $\mathscr{A} = \{u_1, \dots, u_n\}$, $\mathscr{B} = \{v_1, \dots, v_m\}$, $\mathscr{C} = \{w_1, \dots, w_p\}$,

$$\tau(u_i, v_j, w_k) = a_{ijk}$$

- change-of-basis theorem: if

$$[\tau]_{\mathscr{A},\mathscr{B},\mathscr{C}} = A \quad \text{and} \quad [\tau]_{\mathscr{A}',\mathscr{B}',\mathscr{C}'} = A' \in \mathbb{R}^{m\times n\times p},$$

then

$$A' = (X^{\mathsf{T}}, Y^{\mathsf{T}}, Z^{\mathsf{T}}) \cdot A$$

extends to arbitrary order

- bilinear operators $\longleftrightarrow$ mixed 3-tensor of covariant order 2 contravariant order 1
- trilinear functionals $\longleftrightarrow$ covariant 3-tensor
- recover all transformation rules in definition ①
  - covariant d-tensor:

$$A' = (X_1^{\mathsf{T}}, X_2^{\mathsf{T}}, \dots, X_d^{\mathsf{T}}) \cdot A$$

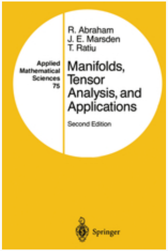
  - contravariant d-tensor:

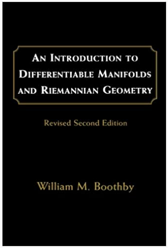
$$A' = (X_1^{-1}, X_2^{-1}, \dots, X_d^{-1}) \cdot A$$

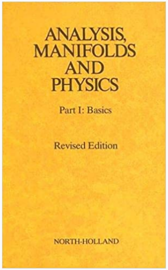
  - mixed d-tensor:

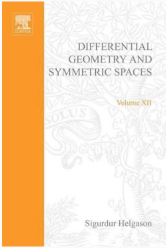
$$A' = (X_1^{-1}, \dots, X_p^{-1}, X_{p+1}^{\mathsf{T}}, \dots, X_d^{\mathsf{T}}) \cdot A$$

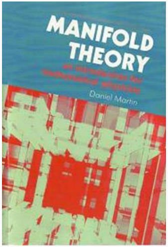
- definition ② is the easiest definition of a tensor

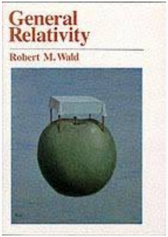

R. Abraham
J. E. Marsden
T. Ratiu
Applied Mathematical Sciences 75
Manifolds, Tensor Analysis, and Applications
Second Edition
Springer

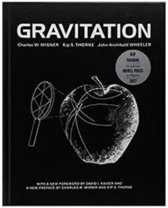

An Introduction to Differentiable Manifolds and Riemannian Geometry
Revised Second Edition
William M. Boothby

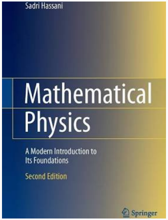

ANALYSIS, MANIFOLDS AND PHYSICS
Part I: Basics
Revised Edition
NORTH-HOLLAND


DIFFERENTIAL GEOMETRY AND SYMMETRIC SPACES
Volume XII
Sigurdur Helgason


MANIFOLD THEORY
Daniel Martin


General Relativity
Robert M. Wald


GRAVITATION
Charles W. MISNER Kip S. THORNE John Archibald WHEELER


Sadri Hassani
Mathematical Physics
A Modern Introduction to Its Foundations
Second Edition
Springer

issue

- many more multilinear maps than there are types of tensors
- $d = 2$:
  - linear operators

$$\Phi : \mathbb{U}^* \to \mathbb{V}, \quad \Phi : \mathbb{U} \to \mathbb{V}^*, \quad \Phi : \mathbb{U}^* \to \mathbb{V}^*$$

  - bilinear functionals

$$\beta : \mathbb{U}^* \times \mathbb{V} \to \mathbb{R}, \quad \beta : \mathbb{U} \times \mathbb{V}^* \to \mathbb{R}, \quad \beta : \mathbb{U}^* \times \mathbb{V}^* \to \mathbb{R}$$

- $d = 3$:
  - bilinear operators

$$\mathrm{B} : \mathbb{U}^* \times \mathbb{V} \to \mathbb{W},\ \mathrm{B} : \mathbb{U} \times \mathbb{V}^* \to \mathbb{W}, \dots, \mathrm{B} : \mathbb{U}^* \times \mathbb{V}^* \to \mathbb{W}^*$$

  - trilinear functionals

$$\tau : \mathbb{U}^* \times \mathbb{V} \times \mathbb{W} \to \mathbb{R},\ \tau : \mathbb{U} \times \mathbb{V}^* \times \mathbb{W} \to \mathbb{R}, \dots, \tau : \mathbb{U}^* \times \mathbb{V}^* \times \mathbb{W}^* \to \mathbb{R}$$

  - more complicated maps

$$\Phi_1 : \mathbb{U} \to \mathsf{L}(\mathbb{V}; \mathbb{W}), \quad \Phi_2 : \mathsf{L}(\mathbb{U}; \mathbb{V}) \to \mathbb{W},$$
$$\beta_1 : \mathbb{U} \times \mathsf{L}(\mathbb{V}; \mathbb{W}) \to \mathbb{R}, \quad \beta_2 : \mathsf{L}(\mathbb{U}; \mathbb{V}) \times \mathbb{W} \to \mathbb{R}$$

issue

- possibilities increase exponentially with order d
- ought to be only as many as types of transformation rules

$$
\begin{aligned}
&\text{covariant 2-tensor:} && \Phi : \mathbb{U} \to \mathbb{V}^*, && \beta : \mathbb{U} \times \mathbb{V} \to \mathbb{R} \\
&\text{contravariant 2-tensor:} && \Phi : \mathbb{U}^* \to \mathbb{V}, && \beta : \mathbb{U}^* \times \mathbb{V}^* \to \mathbb{R} \\
&\text{mixed 2-tensor:} && \Phi : \mathbb{U} \to \mathbb{V}, && \beta : \mathbb{U} \times \mathbb{V}^* \to \mathbb{R}, \\
& && \Phi : \mathbb{U}^* \to \mathbb{V}^*, && \beta : \mathbb{U}^* \times \mathbb{V} \to \mathbb{R}
\end{aligned}
$$

- definition ③ accomplishes this without reference to the transformation rules

imperfect fix

- only allow $\mathbb{W} = \mathbb{R}$
- d-tensor of contravariant order p and covariant order $d - p$ is multilinear functional

$$\varphi : \mathbb{V}_1^* \times \cdots \times \mathbb{V}_p^* \times \mathbb{V}_{p+1} \times \cdots \times \mathbb{V}_d \to \mathbb{R}$$

- excludes vectors, by far the most common 1-tensor
- excludes linear operators, by far the most common 2-tensor
- excludes bilinear operators, by far the most common 3-tensor
- e.g., instead of talking about $v \in \mathbb{V}$, need to talk about linear functionals $f : \mathbb{V}^* \to \mathbb{R}$

multilinear operators are useful

- favorite example: higher derivatives of multivariate functions
- but need a norm on $\mathsf{M}^d(\mathbb{V}_1, \ldots, \mathbb{V}_d; \mathbb{W})$
- $\mathsf{M}^d(\mathbb{V}_1, \ldots, \mathbb{V}_d; \mathbb{W})$ is itself a vector space
- if $\mathbb{V}_1, \ldots, \mathbb{V}_d$ and $\mathbb{W}$ endowed with norms, then

$$\|\Phi\|_\sigma := \sup_{v_1, \ldots, v_d \neq 0} \frac{\|\Phi(v_1, \ldots, v_d)\|}{\|v_1\| \cdots \|v_d\|}$$

  defines a norm on $\mathsf{M}^d(\mathbb{V}_1, \ldots, \mathbb{V}_d; \mathbb{W})$
- slightly abused notation: same $\|\cdot\|$ denote norms on different spaces

higher-order derivatives

- $\mathbb{V}$, $\mathbb{W}$ normed spaces; $\Omega \subseteq \mathbb{V}$ open
- derivative of $f : \Omega \to \mathbb{W}$ at $v \in \Omega$ is linear operator $Df(v) : \mathbb{V} \to \mathbb{W}$,

$$\lim_{h \to 0} \frac{\|f(v+h) - f(v) - [Df(v)](h)\|}{\|h\|} = 0$$

- since $Df(v) \in \mathsf{L}(\mathbb{V}; \mathbb{W})$, apply same definition to $Df : \Omega \to \mathsf{L}(\mathbb{V}; \mathbb{W})$
- get $D^2 f(v) : \mathbb{V} \to \mathsf{L}(\mathbb{V}; \mathbb{W})$ as $D(Df)$,

$$\lim_{h \to 0} \frac{\|Df(v+h) - Df(v) - [D^2 f(v)](h)\|}{\|h\|} = 0$$

- apply recursively to get derivatives of arbitrary order

$$\begin{aligned} Df(v) &\in \mathsf{L}(\mathbb{V}; \mathbb{W}), & D^2 f(v) &\in \mathsf{L}\big(\mathbb{V}; \mathsf{L}(\mathbb{V}; \mathbb{W})\big), \\ D^3 f(v) &\in \mathsf{L}\big(\mathbb{V}; \mathsf{L}(\mathbb{V}; \mathsf{L}(\mathbb{V}; \mathbb{W}))\big), & D^4 f(v) &\in \mathsf{L}\Big(\mathbb{V}; \mathsf{L}\big(\mathbb{V}; \mathsf{L}(\mathbb{V}; \mathsf{L}(\mathbb{V}; \mathbb{W}))\big)\Big) \end{aligned}$$

higher-order derivatives

- how to avoid nested spaces of linear maps?
- use multilinear maps

$$\mathsf{L}\big(\mathbb{V};\mathsf{M}^{d-1}(\mathbb{V},\ldots,\mathbb{V};\mathbb{W})\big) = \mathsf{M}^{d}(\mathbb{V},\ldots,\mathbb{V};\mathbb{W})$$

- if $\Phi : \mathbb{V} \to \mathsf{M}^{d}(\mathbb{V},\ldots,\mathbb{V};\mathbb{W})$ linear, then

$$[\Phi(h)](h_1,\ldots,h_d)$$

  linear in h for fixed $h_1,\ldots,h_d$, d-linear in $h_1,\ldots,h_d$ for fixed h
- $D^d f(v) : \mathbb{V} \times \cdots \times \mathbb{V} \to \mathbb{W}$ may be regarded as multilinear operator
- Taylor's theorem

$$f(v+h) = f(v) + [Df(v)](h) + \frac{1}{2}[D^2 f(v)](h,h) + \cdots$$
$$\cdots + \frac{1}{d!}[D^d f(v)](h,\ldots,h) + R(h)$$

  remainder $\|R(h)\|/\|h\|^d \to 0$ as $h \to 0$

why not hypermatrices

- why not choose bases and just treat them as hypermatrices $A \in \mathbb{R}^{n_1 \times \cdots \times n_d}$?
- may not be computable: writing down a hypermatrix given bases in general #P-hard
- may not be possible: multilinear maps extend to modules, which may not have bases
- may not be useful: even for $d = 1, 2$, writing down matrix unhelpful when vector spaces have special structures
- may not be meaningful: ‘indices’ may be continuous

writing down hypermatrix is #P-hard

- $0 \le d_1 \le d_2 \le \cdots \le d_n$, generalized Vandermonde matrix

$$V_{(d_1,\dots,d_n)}(x) := \begin{bmatrix} x_1^{d_1} & x_2^{d_1} & \dots & x_n^{d_1} \\ x_1^{d_2} & x_2^{d_2} & \dots & x_n^{d_2} \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{d_{n-1}} & x_2^{d_{n-1}} & \dots & x_n^{d_{n-1}} \\ x_1^{d_n} & x_2^{d_n} & \dots & x_n^{d_n} \end{bmatrix}$$

- usual Vandermonde matrix

$$V_{(0,1,\dots,n-1)}(x) = \begin{bmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_n \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{n-2} & x_2^{n-2} & \dots & x_n^{n-2} \\ x_1^{n-1} & x_2^{n-1} & \dots & x_n^{n-1} \end{bmatrix}$$

- if $d_i \ge i$, then $\det V_{(d_1,d_2,\dots,d_n)}(x)$ divisible by $\det V_{(0,1,\dots,n-1)}(x)$

writing down hypermatrix is #P-hard

- for any integers $0 \le p_1 \le p_2 \le \cdots \le p_n$,

$$s_{(p_1,p_2,\ldots,p_n)}(x) := \frac{\det V_{(p_1,p_2+1,\ldots,p_n+n-1)}(x)}{\det V_{(0,1,\ldots,n-1)}(x)}$$

  is a symmetric polynomial in the variables $x_1,\ldots,x_n$

$$s(x_1,x_2,\ldots,x_n) = s\bigl(x_{\sigma(1)},x_{\sigma(2)},\ldots,x_{\sigma(n)}\bigr)$$

- $\mathbb{U}$, $\mathbb{V}$, $\mathbb{W}$ vector spaces of symmetric polynomials of degrees d, d', and $d+d'$ respectively
- $\mathrm{B} : \mathbb{U} \times \mathbb{V} \to \mathbb{W}$ bilinear operator given by

$$\mathrm{B}\bigl(s(x),t(x)\bigr) = s(x)t(x)$$

  for $s(x)$ of degree d and $t(x)$ of degree d'

writing down hypermatrix is #P-hard

- $\mathscr{A}$ be basis of $\mathbb{U}$ given by

$$\left\{s_{(p_1,p_2,\dots,p_n)}(x) \in \mathbb{U} : p_1 \le p_2 \le \cdots \le p_n \text{ integer partition of } d\right\}$$

- $\mathscr{B}$, $\mathscr{C}$ similar bases for $\mathbb{V}$, $\mathbb{W}$
- B may in principle be written down as 3-dimensional hypermatrix

$$[\mathrm{B}]_{\mathscr{A},\mathscr{B},\mathscr{C}} = A \in \mathbb{R}^{d \times d' \times (d+d')}$$

with

$$\mathrm{B}(u_i, v_j) = \sum_{i=1}^{p} a_{ijk} w_k$$

- a_{ijk}'s are Littlewood–Richardson coefficients, well-known to be #P-complete [Narayanan, 2006]
- linear algebra significance: Horn's conjecture on eigenvalues of sums of Hermitian matrices [Klyachko 1998; Knutson–Tao, 1999]

multilinear maps in computations

higher derivatives of log det

- barrier function for positive definite cone $\mathbb{S}^n_{++}$

$$f : \mathbb{S}^n_{++} \to \mathbb{R}, \qquad f(X) = -\log\det X$$

- gradient is

$$\nabla f : \mathbb{S}^n_{++} \to \mathbb{S}^n, \qquad \nabla f(X) = -X^{-1}$$

- Hessian, i.e., $\nabla^2 f \coloneqq D(\nabla f)$, at any $X \in \mathbb{S}^n_{++}$ is linear operator

$$\nabla^2 f(X) : \mathbb{S}^n \to \mathbb{S}^n, \qquad H \mapsto X^{-1} H X^{-1}$$

- standard formulas useless

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{bmatrix}, \qquad \nabla^2 f = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \cdots & \frac{\partial f}{\partial x_n^2} \end{bmatrix}$$

higher derivatives of log det

- writing down (hyper)matrix representations of (multi)linear maps may not be useful even when $d = 1, 2$
- in SDP, vector space is $\mathbb{S}^n$, $n \times n$ real symmetric matrices
- bad idea to identify it with $\mathbb{R}^{n(n+1)/2}$
- what about higher derivatives of $f(X) = -\log\det X$?
- formulas like

$$\nabla^d f = \left[\frac{\partial^d f}{\partial x_i \partial x_j \cdots \partial x_k} \right]_{i,j,\dots,k=1}^n$$

  even less illuminating
- need to view them as multilinear maps

higher derivatives of log det

- write $F = \nabla f$, i.e., $F(X) = -X^{-1}$
- then $DF(X) = \nabla^2 f(X)$ and now we want $D^2 F(X)$
- by earlier discussion, this is bilinear operator

$$D^2 F(X) : \mathbb{S}^n \times \mathbb{S}^n \to \mathbb{S}^n$$

- not hard to show that it is given by

$$(H_1, H_2) \mapsto -X^{-1} H_1 X^{-1} H_2 X^{-1} - X^{-1} H_2 X^{-1} H_1 X^{-1}.$$

- dth derivative is d-linear operator

$$D^d F(X) : \mathbb{S}^n \times \cdots \times \mathbb{S}^n \to \mathbb{S}^n$$

  that sends $(H_1, H_2, \ldots, H_d)$ to

$$(-1)^{d+1} \sum_{\sigma \in \mathfrak{S}_d} X^{-1} H_{\sigma(1)} X^{-1} H_{\sigma(2)} X^{-1} \cdots X^{-1} H_{\sigma(d)} X^{-1}$$

self-concordance

- need third derivative to check self-concordance
- convex $f : \Omega \subseteq \mathbb{R}^n \to \mathbb{R}$ self-concordant at $x \in \Omega$ if

$$\left|\nabla^3 f(x)(h, h, h)\right| \leq 2\sigma \left|\nabla^2 f(x)(h, h)\right|^{3/2}$$

  for all $h \in \mathbb{R}^n$ [Nesterov–Nemirovskii, 1994]
- convex programming problem may be solved to ε-accuracy in polynomial time if it has self-concordant barrier functions,
- e.g., LP, QP, SOCP, SDP, GP

- not useful when vector space is not $\mathbb{R}^n$:

$$\nabla^2 f(x)(h,h) = \sum_{i,j=1}^{n} \frac{\partial^2 f(x)}{\partial x_i \partial x_j} h_i h_j,$$
$$\nabla^3 f(x)(h,h,h) = \sum_{i,j,k=1}^{n} \frac{\partial^3 f(x)}{\partial x_i \partial x_j \partial x_k} h_i h_j h_k$$

- to check that $f(X) = -\log\det(X)$ is self-concordant, need to show

$$\left|\operatorname{tr}\left(H^\intercal[\nabla^3 f(X)](H,H)\right)\right| \le 2\sigma \left|\operatorname{tr}\left(H^\intercal[\nabla^2 f(X)](H)\right)\right|^{3/2}$$

- easy with multilinear map formulas

$$[\nabla^2 f(X)](H) = X^{-1} H X^{-1},$$
$$[\nabla^3 f(X)](H,H) = -2X^{-1} H X^{-1} H X^{-1}$$

bilinear complexity

- given $\mathbb{U}$, $\mathbb{V}$, $\mathbb{W}$, how to construct a bilinear operator

$$\mathrm{B} : \mathbb{U} \times \mathbb{V} \to \mathbb{W}?$$

- take linear functional $\varphi : \mathbb{U} \to \mathbb{R}$, linear functional $\psi : \mathbb{V} \to \mathbb{R}$, vector $w \in \mathbb{W}$, define

$$\mathrm{B}(u, v) = \varphi(u)\psi(v)w$$

for any $u \in \mathbb{U}$, $v \in \mathbb{V}$

- evaluating B requires exactly one multiplication of variables

bilinear complexity

- for example $\mathbb{U} = \mathbb{V} = \mathbb{W} = \mathbb{R}^3$ with

$$\varphi(u) = u_1 + 2u_2 + 3u_3,$$
$$\psi(v) = 2v_1 + 3v_2 + 4v_3,$$
$$w = (3, 4, 5)$$

  then

$$\mathrm{B}(u, v) = \begin{bmatrix} 3(u_1 + 2u_2 + 3u_3)(2v_1 + 3v_2 + 4v_3) \\ 4(u_1 + 2u_2 + 3u_3)(2v_1 + 3v_2 + 4v_3) \\ 5(u_1 + 2u_2 + 3u_3)(2v_1 + 3v_2 + 4v_3) \end{bmatrix}$$

- multiplications like $2u_2$ or $4v_3$ are all scalar multiplications, i.e., one of the factors is a constant
- only variable multiplication like $(u_1 + 2u_2 + 3u_3)(2v_1 + 3v_2 + 4v_3)$ counts

bilinear complexity

- this is the notion of bilinear complexity [Strassen, 1987]
- once we fixed φ, ψ, w, evaluation of these can be hardwired or hardcoded
- e.g., discrete Fourier transform

$$\begin{bmatrix} x'_0 \\ x'_1 \\ x'_2 \\ x'_3 \\ \vdots \\ x'_{n-1} \end{bmatrix} = \frac{1}{\sqrt{n}} \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \omega^3 & \cdots & \omega^{n-1} \\ 1 & \omega^2 & \omega^4 & \omega^6 & \cdots & \omega^{2(n-1)} \\ 1 & \omega^3 & \omega^6 & \omega^9 & \cdots & \omega^{3(n-1)} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{n-1} & \omega^{2(n-1)} & \omega^{3(n-1)} & \cdots & \omega^{(n-1)(n-1)} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{n-1} \end{bmatrix}$$

- use FFT to evaluate $\varphi(x) = \sum_{j=0}^{n-1} \omega^{jk} x_j$
- bilinear complexity of FFT is zero

bilinear complexity

- may often bound number of additions and scalar multiplications in terms of number of variable multiplications
- e.g., if an algorithm takes n^p variable multiplications, may show that it takes at most $10n^p$ additions and scalar multiplications
- so algorithm still $O(n^p)$ even if we count all arithmetic operations
- most importantly, **bilinear complexity = tensor rank**

$$\operatorname{rank}(\mathrm{B}) = \min\left\{ r : \mathrm{B}(u,v) = \sum_{i=1}^{r} \varphi_i(u)\psi_i(v)w_i \right\}$$

- if only need $\mathrm{B}(u,v)$ up to ε-accuracy, **border rank**

$$\overline{\operatorname{rank}}(\mathrm{B}) = \min\left\{ r : \mathrm{B}(u,v) = \lim_{\varepsilon \to 0^+} \sum_{i=1}^{r} \varphi_i^\varepsilon(u)\psi_i^\varepsilon(v)w_i^\varepsilon \right\}$$

- due to [Strassen, 1973] and [Bini–Lotti–Romani, 1980] respectively

Gauss's algorithm

- complex multiplication with three real multiplications

$$
\begin{aligned}
(a+bi)(c+di) &= (ac-bd)+i(bc+ad) \\
&= (ac-bd)+i[(a+b)(c+d)-ac-bd]
\end{aligned}
$$

- $\mathrm{B} : \mathbb{C} \times \mathbb{C} \to \mathbb{C}$, $(z, w) \mapsto zw$ is $\mathbb{R}$-bilinear

$$
\mathrm{B} : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}^2, \quad \mathrm{B}\left(\begin{bmatrix} a \\ b \end{bmatrix}, \begin{bmatrix} c \\ d \end{bmatrix} \right) = \begin{bmatrix} ac-bd \\ bc+ad \end{bmatrix}
$$

- usual:

$$
\mathrm{B}(z,w) = [e_1^*(z)e_1^*(w) - e_2^*(z)e_2^*(w)]e_1 + [e_1^*(z)e_2^*(w) + e_2^*(z)e_1^*(w)]e_2
$$

- Gauss:

$$
\begin{aligned}
\mathrm{B}(z,w) = {} & [(e_1^* + e_2^*)(z)(e_1^* + e_2^*)(w)]e_2 \\
& + [e_1^*(z)e_1^*(w)](e_1 - e_2) - [e_2^*(z)e_2^*(w)](e_1 + e_2)
\end{aligned}
$$

Gauss's algorithm

- Gauss optimal in both exact and approximate sense:

$$\operatorname{rank}(\mathrm{B}) = 3 = \overline{\operatorname{rank}}(\mathrm{B})$$

- why useful?
- complex matrix multiplication:

$$(A + iB)(C + iD) = (AC - BD) + i[(A + B)(C + D) - AC - BD]$$

for $A + iB, C + iD \in \mathbb{C}^{n\times n}$ with $A, B, C, D \in \mathbb{R}^{n\times n}$

- which is why we should allow for modules
    - $\mathbb{C}$ two-dimensional vector space over $\mathbb{R}$
    - $\mathbb{C}^{n\times n}$ two-dimensional free module over $\mathbb{R}^{n\times n}$

other simple example?

- Gauss essentially the only one in two dimensions
- parallel evaluation of standard inner product and standard symplectic form on $\mathbb{R}^2$

$$g(x, y) = x_1 y_1 + x_2 y_2 \quad \text{and} \quad \omega(x, y) = x_1 y_2 - x_2 y_1.$$

- algorithm similar to Gauss's gives result with $\operatorname{rank}(\mathrm{B}) = 3 = \overline{\operatorname{rank}}(\mathrm{B})$
- three dimensions: skew-symmetric matrix-vector product

$$\begin{bmatrix} 0 & a & b \\ -a & 0 & c \\ -b & -c & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} ay + bz \\ -ax + cz \\ -bx - cy \end{bmatrix}$$

- in this case[1] $\operatorname{rank}(\mathrm{B}) = 5 = \overline{\operatorname{rank}}(\mathrm{B})$

[1] thanks to J. M. Landsberg (for $\mathbb{C}$) and Visu Makam (for $\mathbb{R}$)

Strassen's algorithm

- 2×2 matrix multiplication with seven multiplications

$$\begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix} \begin{bmatrix} b_1 & b_2 \\ b_3 & b_4 \end{bmatrix} = \begin{bmatrix} a_1 b_1 + a_2 b_2 & \beta + \gamma + (a_1 + a_2 - a_3 - a_4) b_4 \\ \alpha + \gamma + a_4 (b_2 + b_3 - b_1 - b_4) & \alpha + \beta + \gamma \end{bmatrix}$$

with

$$\alpha = (a_3 - a_1)(b_3 - b_4),\ \beta = (a_3 + a_4)(b_3 - b_1),\ \gamma = a_1 b_1 + (a_3 + a_4 - a_1)(b_1 + b_4 - b_3)$$

- consequence: inverting $n \times n$ matrix in $5.64 n^{\log_2 7}$ arithmetic operations (both additions and multiplications) [Strassen, 1969]
- huge surprise as there were results showing $n^3/3$ required by Gaussian elimination cannot be improved
- such results assume row and column operations, Strassen used block operations
- $\operatorname{rank}(\mathrm{B}) = 7 = \overline{\operatorname{rank}}(\mathrm{B})$ [Landsberg, 2006]

matrix multiplication tensor

- bilinear operator

$$\mu_{m,n,p} : \mathbb{R}^{m\times n} \times \mathbb{R}^{n\times p} \to \mathbb{R}^{n\times p}, \quad (A, B) \mapsto AB$$

  called matrix multiplication tensor

- exponent of matrix multiplication is

$$\omega := \inf\{p \in \mathbb{R} : \operatorname{rank}(\mu_{n,n,n}) = O(n^p)\}$$

- current bound $\omega < 2.3728596$ [Alman–Vassilevska Williams, 2021]
- ω underlies nearly every problem in numerical linear algebra

exponent of matrix multiplication

- inversion: given $A \in \mathrm{GL}(n)$, find $A^{-1} \in \mathrm{GL}(n)$
- determinant: given $A \in \mathrm{GL}(n)$, find $\det(A) \in \mathbb{R}$
- null basis: given $A \in \mathbb{R}^{n \times n}$, find a basis $v_1, \ldots, v_m \in \mathbb{R}^n$ of $\ker(A)$
- linear system: given $A \in \mathrm{GL}(n)$ and $b \in \mathbb{R}^n$, find $v \in \mathbb{R}^n$ so that $Av = b$
- LU decomposition: given $A \in \mathbb{R}^{m \times n}$ of full rank, find permutation P, unit lower triangular $L \in \mathbb{R}^{m \times m}$, upper triangular $U \in \mathbb{R}^{m \times n}$ so that $PA = LU$
- QR decomposition: given $A \in \mathbb{R}^{n \times n}$, find orthogonal $Q \in \mathrm{O}(n)$, upper triangular $U \in \mathbb{R}^{n \times n}$ so that $A = QR$

exponent of matrix multiplication

- eigenvalue decomposition: given $A \in \mathbb{S}^n$, find $Q \in \mathrm{O}(n)$ and diagonal $\Lambda \in \mathbb{R}^{n \times n}$ so that $A = Q\Lambda Q^{\mathsf{T}}$
- Hessenberg decomposition: given $A \in \mathbb{R}^{n \times n}$, find $Q \in \mathrm{O}(n)$ and upper Hessenberg $H \in \mathbb{R}^{n \times n}$ so that $A = QHQ^{\mathsf{T}}$
- characteristic polynomial: given $A \in \mathbb{R}^{n \times n}$, find $(a_0, \dots, a_{n-1}) \in \mathbb{R}^n$ so that $\det(xI - A) = x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$
- sparsification: given $A \in \mathbb{R}^{n \times n}$ and $c \in [1, \infty)$, find $X, Y \in \mathrm{GL}(n)$ so that $\mathrm{nnz}(XAY^{-1}) \le cn$

exponent of nearly all matrix computations

any $\varepsilon > 0$, there is an algorithm for each of these problems in $O(n^{\omega+\varepsilon})$ arithmetic operations (including additions and scalar multiplications)

thanks

- my former postdocs Ke Ye (Chinese Academy of Sciences), Yang Qi (École Polytechnique and INRIA): taught me everything about tensors
- Keith Conrad, Shmuel Friedland, Edinah Gnang, Shenglong Hu, Risi Kondor, J. M. Landsberg, Jiawang Nie, Peter McCullagh, Emily Riehl, Thomas Schultz: learned a great deal from them too
- image credits: Sanpaz (p. 3, Cauchy stress), Waveguide2 (p. 6, Ohm's law), svjo (p. 6, Hooke's law)