From Matrix to Tensor: The Transition to Numerical Multilinear Algebra

Lecture 1. Introduction to Tensor Computations

Charles F. Van Loan
Cornell University

The Gene Golub SIAM Summer School 2010
Selva di Fasano, Brindisi, Italy
What is a Tensor?

Definition

An order-\(d\) tensor \(\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}\) is a real \(d\)-dimensional array \(\mathcal{A}(1:n_1, \ldots, 1:n_d)\) where the index range in the \(k\)-th mode is from 1 to \(n_k\).

Low-Order Tensors

A scalar is a \(0\)-order tensor.
A vector is an order-1 tensor.
A matrix is an order-2 tensor.

We will use calligraphic font to designate tensors that have order 3 or greater, e.g., \(\mathcal{A}, \mathcal{B}, \mathcal{C}\), etc.
Discretization.

\( A(i, j, k, \ell) \) might house the value of \( f(w, x, y, z) \) at 
\((w, x, y, z) = (w_i, x_j, y_k, z_\ell)\).

Multiway Analysis.

\( A(i, j, k, \ell) \) is a value that captures an interaction between four variables/factors.
A color picture is an $m$-by-$n$-by-3 tensor:

$A(:,:,1) = \text{red pixel values}$

$A(:,:,2) = \text{green pixel values}$

$A(:,:,3) = \text{blue pixel values}$
You Have Seen them Before

Block Matrices

\[
A = \begin{bmatrix}
  a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} \\
  a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} \\
  a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & a_{36} \\
  a_{41} & a_{42} & a_{43} & a_{44} & a_{45} & a_{46} \\
  a_{51} & a_{52} & a_{53} & a_{54} & a_{55} & a_{56} \\
  a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & a_{66}
\end{bmatrix}
\]

Matrix entry \( a_{45} \) is the \((2,1)\) entry of the \((2,3)\) block:

\[
a_{45} \iff A(2, 3, 2, 1)
\]
Typical:

Convert the given problem into an equivalent easy-to-solve problem by using the “right” matrix decomposition.

\[ PA = LU, \quad Ly = Pb, \quad Ux = y \quad \Rightarrow \quad Ax = b \]

Also Typical:

Uncover hidden relationships by computing the “right” decomposition of the data matrix.

\[ A = U\Sigma V^T \quad \Rightarrow \quad A \approx \sum_{i=1}^{\hat{r}} \sigma_i u_i v_i^T \]
Two Natural Questions

Question 1.
Can we solve tensor problems by converting them to equivalent, easy-to-solve problems using a tensor decomposition?

Question 2.
Can we uncover hidden patterns in tensor data by computing an appropriate tensor decomposition?

We explore the decomposition issue for 2-by-2-by-2 tensors...
The Singular Value Decomposition

The 2-by-2 case...

\[
\begin{bmatrix}
a_{11} & a_{12} \\
a_{21} & a_{22}
\end{bmatrix}
= 
\begin{bmatrix}
u_{11} & u_{12} \\
u_{21} & u_{22}
\end{bmatrix}
\begin{bmatrix}
\sigma_1 & 0 \\
0 & \sigma_2
\end{bmatrix}
\begin{bmatrix}
v_{11} & v_{12} \\
v_{21} & v_{22}
\end{bmatrix}^T
\]

\[
= \sigma_1
\begin{bmatrix}
u_{11} \\
u_{21}
\end{bmatrix}
\begin{bmatrix}
v_{11}
\end{bmatrix}^T
+ \sigma_2
\begin{bmatrix}
u_{12} \\
u_{22}
\end{bmatrix}
\begin{bmatrix}
v_{12}
\end{bmatrix}^T
\]

A reshaped presentation of the same thing...

\[
\begin{bmatrix}
a_{11} \\
a_{21} \\
a_{12} \\
a_{22}
\end{bmatrix}
= \sigma_1
\begin{bmatrix}
v_{11}u_{11} \\
v_{11}u_{21} \\
v_{21}u_{11} \\
v_{21}u_{21}
\end{bmatrix}
+ \sigma_2
\begin{bmatrix}
v_{12}u_{12} \\
v_{12}u_{22} \\
v_{22}u_{12} \\
v_{22}u_{22}
\end{bmatrix}
\]
The Kronecker Product

Definition As Applied to 2-vectors

\[
\begin{bmatrix}
x_1 \\
x_2 \\
\end{bmatrix} \otimes \begin{bmatrix}
y_1 \\
y_2 \\
\end{bmatrix} = \begin{bmatrix}
x_1 y_1 \\
x_1 y_2 \\
x_2 y_1 \\
x_2 y_2 \\
\end{bmatrix}
\]

2-by-2 SVD in Kronecker Terms...

\[
\begin{bmatrix}
a_{11} \\
a_{21} \\
a_{12} \\
a_{22} \\
\end{bmatrix} = \sigma_1 \begin{bmatrix}
v_{11} \\
v_{21} \\
\end{bmatrix} \otimes \begin{bmatrix}
u_{11} \\
u_{21} \\
\end{bmatrix} + \sigma_2 \begin{bmatrix}
v_{12} \\
v_{22} \\
\end{bmatrix} \otimes \begin{bmatrix}
u_{12} \\
u_{22} \\
\end{bmatrix}
\]
Appreciate the Duality..

\[
\begin{bmatrix}
a_{11} & a_{12} \\
a_{21} & a_{22}
\end{bmatrix}
= \sigma_1 \cdot u_1 v_1^T + \sigma_2 \cdot u_2 v_2^T
\]

\[
\begin{bmatrix}
a_{11} \\
a_{21} \\
a_{12} \\
a_{22}
\end{bmatrix}
= \sigma_1 \cdot (v_1 \otimes u_1) + \sigma_2 \cdot (v_2 \otimes u_1)
\]

\[
U = \begin{bmatrix}
u_1 \\
u_2
\end{bmatrix} \quad V = \begin{bmatrix}
v_1 \\
v_2
\end{bmatrix}
\]
Write $A \in \mathbb{R}^{2 \times 2 \times 2}$ as a minimal sum of rank-1 tensors.

**What’s a rank-1 tensor?**

If $R$ has unit rank then

$$R = (r_{ijk}) \quad r_{ijk} = h_k g_j f_i$$

If $R \in \mathbb{R}^{2 \times 2 \times 2}$ has unit rank then there exist $f, g, h \in \mathbb{R}^2$ such that

$$\begin{bmatrix}
  r_{111} \\
  r_{211} \\
  r_{121} \\
  r_{221} \\
  r_{112} \\
  r_{212} \\
  r_{122} \\
  r_{222}
\end{bmatrix} = \begin{bmatrix}
  h_1 \\
  h_2
\end{bmatrix} \otimes \begin{bmatrix}
  g_1 \\
  g_2
\end{bmatrix} \otimes \begin{bmatrix}
  f_1 \\
  f_2
\end{bmatrix}$$
Write $\mathcal{A} \in \mathbb{R}^{2 \times 2 \times 2}$ as a minimal sum of rank-1 tensors.

Find thinnest possible $X, Y, Z \in \mathbb{R}^{2 \times r}$ so

$$
\begin{bmatrix}
a_{111} \\
a_{211} \\
a_{121} \\
a_{221} \\
a_{112} \\
a_{212} \\
a_{122} \\
a_{222}
\end{bmatrix}
= \sum_{k=1}^{r} z_k \otimes y_k \otimes x_k
$$

where

$$X = [x_1|\cdots|x_r] \quad Y = [y_1|\cdots|y_r] \quad Z = [z_1|\cdots|z_r]$$
Write $\mathcal{A} \in \mathbb{R}^{2 \times 2 \times 2}$ as a minimal sum of rank-1 tensors.

A Surprising Fact

If

\[
\begin{bmatrix}
a_{111} \\
a_{211} \\
a_{121} \\
a_{221} \\
a_{112} \\
a_{212} \\
a_{122} \\
a_{222}
\end{bmatrix} = \text{randn}(8, 1)
\]

then

\[
\begin{cases}
r = 2 & 79\% \text{ of the time} \\
r = 3 & 21\% \text{ of the time}
\end{cases}
\]

Compare to..

If $A = \text{randn}(n, n)$, then 100% of the time $\text{rank}(A) = n$

What are the “full rank” 2-by-2-by-2 tensors?
The 79/21 Property

Interesting Fact

If the $a_{ijk}$ are randn then

$$\det \left( \begin{bmatrix} a_{111} & a_{121} \\ a_{211} & a_{221} \end{bmatrix} - \lambda \begin{bmatrix} a_{112} & a_{122} \\ a_{212} & a_{222} \end{bmatrix} \right) = 0$$

has real distinct roots 79% of the time and complex conjugate roots 21% of the time.

What is the connection between this generalized eigenvalue problem and the 2-by-2-by-2 tensor rank problem?
The 79% Situation

(Real Distinct Eigenvalues)

There exist nonsingular $X = \begin{bmatrix} x_1 & x_2 \end{bmatrix}$ and $Y = \begin{bmatrix} y_1 & y_2 \end{bmatrix}$ so

$$\begin{bmatrix} a_{111} & a_{121} \\ a_{211} & a_{221} \end{bmatrix} = X \begin{bmatrix} \alpha_1 & 0 \\ 0 & \alpha_2 \end{bmatrix} Y^T = \alpha_1 x_1 y_1^T + \alpha_2 x_2 y_2^T$$

$$\begin{bmatrix} a_{112} & a_{122} \\ a_{212} & a_{222} \end{bmatrix} = X \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} Y^T = x_1 y_1^T + x_2 y_2^T$$

i.e.,

$$\begin{bmatrix} a_{111} \\ a_{211} \\ a_{121} \\ a_{221} \end{bmatrix} = \alpha_1 (y_1 \otimes x_1) + \alpha_2 (y_2 \otimes x_2)$$

$$\begin{bmatrix} a_{112} \\ a_{212} \\ a_{122} \\ a_{222} \end{bmatrix} = (y_1 \otimes x_1) + (y_2 \otimes x_2)$$
The 79% Situation

Stack 'em

\[
\begin{bmatrix}
a_{111} \\ a_{211} \\ a_{121} \\ a_{221} \\ a_{112} \\ a_{212} \\ a_{122} \\ a_{222}
\end{bmatrix} = \begin{bmatrix}
\alpha_1 \\ 1
\end{bmatrix} \otimes (y_1 \otimes x_1) + \begin{bmatrix}
\alpha_2 \\ 1
\end{bmatrix} \otimes (y_2 \otimes x_2)
\]

\(A\) has rank 2.
(Complex Conjugate Eigenvalues)

There exist nonsingular $X = \begin{bmatrix} x_1 & x_2 \end{bmatrix}$ and $Y = \begin{bmatrix} y_1 & y_2 \end{bmatrix}$ so

$$\begin{bmatrix} a_{111} & a_{121} \\ a_{211} & a_{221} \end{bmatrix} = X \begin{bmatrix} \alpha_1 & \alpha_2 \\ -\alpha_2 & \alpha_1 \end{bmatrix} Y^T$$

$$= \alpha_1(x_1y_1^T + x_2y_2^T) + \alpha_2(x_1y_2^T - x_2y_1^T)$$

$$= \alpha_1(x_1 + x_2)(y_1 + y_2)^T - (\alpha_1 - \alpha_2)x_1y_2^T - (\alpha_1 + \alpha_2)x_2y_1^T$$

$$\begin{bmatrix} a_{112} & a_{122} \\ a_{212} & a_{222} \end{bmatrix} = X \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} Y^T$$

$$= x_1y_1^T + x_2y_2^T$$

$$= (x_1 + x_2)(y_1 + y_2)^T - x_1y_2^T - x_2y_1^T$$
The 21% Situation

(Complex Conjugate Eigenvalues)

\[
\begin{bmatrix}
  a_{111} \\
  a_{211} \\
  a_{121} \\
  a_{221} \\
  a_{112} \\
  a_{212} \\
  a_{122} \\
  a_{222}
\end{bmatrix}
\begin{bmatrix}
  \alpha_1 \\
  1
\end{bmatrix} 
\otimes (y_1 + y_2) \otimes (x_1 + x_2)

= 

\begin{bmatrix}
  \alpha_2 - \alpha_1 \\
  1
\end{bmatrix} \otimes y_2 \otimes x_1

- 

\begin{bmatrix}
  \alpha_1 + \alpha_2 \\
  1
\end{bmatrix} \otimes y_1 \otimes x_1

\mathcal{A} \text{ has rank 3.}
MATLAB: 2-by-2-by-2 Tensor Rank

% Script L1
% Estimates the prob that randn(2,2,2) has rank 2:

N = 1000000; count = 0;
for eg=1:N
    A = randn(2,2,2);
    A_front = [A(1,1,1) A(1,2,1); A(2,1,1) A(2,2,1)];
    A_back = [A(1,1,2) A(1,2,2); A(2,1,2) A(2,2,2)];
    [S,T] = qz(A_front,A_back,’real’);
    if S(2,1)==0
        count = count+1;
    end
end
ProbARank2 = count/N
Problem 1.1. What happens if we use `rand` instead of `randn` in the script `L1`?

Problem 1.2. Explain why the script `L1` generates the same results regardless of the choice for `A_front` and `A_back`:

Choice 1: $A_{\text{front}} = \begin{bmatrix} a_{111} & a_{121} \\ a_{211} & a_{221} \end{bmatrix}$, $A_{\text{back}} = \begin{bmatrix} a_{112} & a_{122} \\ a_{212} & a_{222} \end{bmatrix}$

Choice 2: $A_{\text{front}} = \begin{bmatrix} a_{111} & a_{112} \\ a_{211} & a_{212} \end{bmatrix}$, $A_{\text{back}} = \begin{bmatrix} a_{121} & a_{122} \\ a_{221} & a_{222} \end{bmatrix}$

Choice 3: $A_{\text{front}} = \begin{bmatrix} a_{111} & a_{112} \\ a_{121} & a_{122} \end{bmatrix}$, $A_{\text{back}} = \begin{bmatrix} a_{211} & a_{212} \\ a_{221} & a_{222} \end{bmatrix}$
**MATLAB: Generating Tensors**

```
% A 2-by-3-by-4 random tensor...
A = randn(2,3,4);

% Dimensions can be specified by a vector...
n = [2,3,4];
A = randn(n);

% The functions ones and zeros...
A = ones(n);
A = zeros(n);

% Size and reshaping...
n = size(A);
N = prod(n);
a = reshape(A,N,1);
B = reshape(a,n(length(n):-1:1));
```
% Applying a function to each component...
\n\n% In lieu of the triple loop...
\n% The usual cast of characters...
\n\nsin(A)
\nfloor(A) ceil(A) round(A) real(A) imag(A) sqrt(A) abs(A)
MATLAB: Tensor Operations

n = [2,3,4];
A = randn(n);
B = randn(n);

% These operations are legal...
C = 3*A;
C = -A;
C = A+1;
C = A.^2;

% These operations are legal if A and B
% have the same dimension...
C = A + B;
C = A./B;
C = A.*B;
C = A.^B;
Nearness Problems

The Nearest Rank-1 Matrix Problem

If \( A \in \mathbb{R}^{n_1 \times n_2} \) has SVD \( A = U\Sigma V^T \) then \( B_{opt} = \sigma_1 u_1 v_1^T \) minimizes

\[
\phi(B) = \| A - B \|_F \quad \text{rank}(B) = 1.
\]

The Nearest Rank-1 Tensor to \( A \in \mathbb{R}^{2 \times 2 \times 2} \)

Find unit 2-norm vectors \( u, v, w \in \mathbb{R}^2 \) and scalar \( \sigma \) so that \( \| a - \sigma \cdot w \otimes v \otimes u \|_2 \) is minimized where

\[
a = \begin{bmatrix}
a_{111} \\
a_{211} \\
a_{112} \\
a_{121} \\
a_{221} \\
a_{122} \\
a_{212} \\
a_{222}
\end{bmatrix}
\]
A Highly Structured Nonlinear Optimization Problem

It depends on four parameters...

\[ \phi(\sigma, \theta_1, \theta_2, \theta_3) = \| a - \sigma \left[ \begin{array}{c} \cos(\theta_3) \\ \sin(\theta_3) \end{array} \right] \otimes \left[ \begin{array}{c} \cos(\theta_2) \\ \sin(\theta_2) \end{array} \right] \otimes \left[ \begin{array}{c} \cos(\theta_1) \\ \sin(\theta_1) \end{array} \right] \|_2 \]

\[
= \left[ \begin{array}{c} a_{111} \\ a_{211} \\ a_{121} \\ a_{221} \\ a_{112} \\ a_{212} \\ a_{122} \\ a_{222} \end{array} \right] - \sigma \cdot \left[ \begin{array}{c} c_3 c_2 c_1 \\ c_3 c_2 s_1 \\ c_3 s_2 c_1 \\ c_3 s_2 s_1 \\ s_3 c_2 c_1 \\ s_3 c_2 s_1 \\ s_3 s_2 c_1 \\ s_3 s_2 s_1 \end{array} \right]_2
\]

\[ c_i = \cos(\theta_i), \ s_i = \sin(\theta_i) \quad i = 1:3 \]
A Highly Structured Nonlinear Optimization Problem

and it can be reshaped...

\[
\phi = \begin{bmatrix}
a_{111} \\
a_{211} \\
a_{121} \\
a_{221} \\
a_{112} \\
a_{212} \\
a_{122} \\
a_{222}
\end{bmatrix} - \sigma \cdot \begin{bmatrix}
c_{3}c_{2}c_{1} \\
c_{3}c_{2}s_{1} \\
c_{3}s_{2}c_{1} \\
c_{3}s_{2}s_{1} \\
s_{3}c_{2}c_{1} \\
s_{3}c_{2}s_{1} \\
s_{3}s_{2}c_{1} \\
s_{3}s_{2}s_{1}
\end{bmatrix} = \begin{bmatrix}
a_{111} \\
a_{211} \\
a_{121} \\
a_{221} \\
a_{112} \\
a_{212} \\
a_{122} \\
a_{222}
\end{bmatrix} - \begin{bmatrix}
c_{3}c_{2} & 0 \\
0 & c_{3}c_{2} \\
c_{3}s_{2} & 0 \\
0 & c_{3}s_{2} \\
s_{3}c_{2} & 0 \\
0 & s_{3}c_{2} \\
s_{3}s_{2} & 0 \\
0 & s_{3}s_{2}
\end{bmatrix} \begin{bmatrix}
x_{1} \\
y_{1}
\end{bmatrix}
\]

\[
\begin{bmatrix}
x_{1} \\
y_{1}
\end{bmatrix} = \sigma \cdot \begin{bmatrix}
c_{1} \\
s_{1}
\end{bmatrix} = \sigma \cdot \begin{bmatrix}
\cos(\theta_{1}) \\
\sin(\theta_{1})
\end{bmatrix}
\]

Idea: Improve \(\sigma\) and \(\theta_{1}\) by minimizing with respect to \(x_{1}\) and \(y_{1}\), holding \(\theta_{2}\) and \(\theta_{3}\) fixed.
A Highly Structured Nonlinear Optimization Problem

and it can be reshaped...

\[
\phi = \begin{bmatrix}
    a_{111} \\
    a_{211} \\
    a_{121} \\
    a_{221} \\
    a_{112} \\
    a_{212} \\
    a_{122} \\
    a_{222}
\end{bmatrix} - \sigma \cdot \begin{bmatrix}
    c_3 c_2 c_1 \\
    c_3 c_2 s_1 \\
    c_3 s_2 c_1 \\
    c_3 s_2 s_1 \\
    s_3 c_2 c_1 \\
    s_3 c_2 s_1 \\
    s_3 s_2 c_1 \\
    s_3 s_2 s_1
\end{bmatrix}_{2} = \begin{bmatrix}
    a_{111} \\
    a_{211} \\
    a_{121} \\
    a_{221} \\
    a_{112} \\
    a_{212} \\
    a_{122} \\
    a_{222}
\end{bmatrix} - \begin{bmatrix}
    c_3 c_1 & 0 \\
    c_3 s_1 & 0 \\
    0 & c_3 c_1 \\
    0 & c_3 s_1 \\
    s_3 c_1 & 0 \\
    s_3 s_1 & 0 \\
    0 & s_3 c_1 \\
    0 & s_3 s_1
\end{bmatrix}_{2} \begin{bmatrix}
    x_2 \\
    y_2
\end{bmatrix}
\]

\[
\begin{bmatrix}
    x_2 \\
    y_2
\end{bmatrix} = \sigma \cdot \begin{bmatrix}
    c_2 \\
    s_2
\end{bmatrix} = \sigma \cdot \begin{bmatrix}
    \cos(\theta_2) \\
    \sin(\theta_2)
\end{bmatrix}
\]

Idea: Improve \( \sigma \) and \( \theta_2 \) by minimizing with respect to \( x_2 \) and \( y_2 \), holding \( \theta_1 \) and \( \theta_3 \) fixed.
A Highly Structured Nonlinear Optimization Problem

and it can be reshaped...

\[
\phi = \begin{bmatrix}
a_{111} \\
a_{211} \\
a_{121} \\
a_{112} \\
a_{212} \\
a_{122} \\
a_{222}
\end{bmatrix} - \sigma \cdot \begin{bmatrix}
c_3 c_2 c_1 \\
c_3 c_2 s_1 \\
c_3 s_2 c_1 \\
s_3 c_2 c_1 \\
s_3 c_2 s_1 \\
s_3 s_2 c_1 \\
s_3 s_2 s_1
\end{bmatrix}_2 = \begin{bmatrix}
a_{111} \\
a_{211} \\
a_{121} \\
a_{112} \\
a_{212} \\
a_{122} \\
a_{222}
\end{bmatrix} - \begin{bmatrix}
c_2 c_1 & 0 \\
c_2 s_1 & 0 \\
s_2 c_1 & 0 \\
0 & c_2 s_1 \\
0 & c_2 s_1 \\
0 & s_2 c_1 \\
0 & s_2 s_1
\end{bmatrix}_2 \begin{bmatrix}
x_3 \\
y_3
\end{bmatrix}
\]

\[
\begin{bmatrix}
x_3 \\
y_3
\end{bmatrix} = \sigma \cdot \begin{bmatrix}
c_3 \\
s_3
\end{bmatrix} = \sigma \cdot \begin{bmatrix}
\cos(\theta_3) \\
\sin(\theta_3)
\end{bmatrix}
\]

Idea: Improve \( \sigma \) and \( \theta_3 \) by minimizing with respect to \( x_3 \) and \( y_3 \), holding \( \theta_1 \) and \( \theta_2 \) fixed.
A Common Framework for Tensor-Related Optimization

- Choose a subset of the unknowns such that if they are (temporarily) fixed, then we are presented with an easy-to-solve problem in the remaining unknowns.

- By choosing different subsets, cycle through all the unknowns.

- Repeat until converged.

“Easy-to-solve” usually means “linear.”
Problem 1.3. Write a MATLAB function \([\sigma, \theta] = \text{NearestTank1}(A)\) that takes a tensor \(A \in \mathbb{R}^{2 \times 2 \times 2}\) and uses alternating least squares to produce an estimate of a nearest rank-1 approximant. What is a good starting value? The linear least squares problems that need to be solved are highly structured. Explain and exploit.

Problem 1.4. Fix \(\theta_3\). How would you choose \(\sigma, \theta_1\) and \(\theta_2\) so that

\[
\phi(\sigma, \theta_1, \theta_2, \theta_3) = \|a - \sigma \begin{bmatrix} \cos(\theta_3) \\ \sin(\theta_3) \end{bmatrix} \otimes \begin{bmatrix} \cos(\theta_2) \\ \sin(\theta_2) \end{bmatrix} \otimes \begin{bmatrix} \cos(\theta_1) \\ \sin(\theta_1) \end{bmatrix}\|_2
\]

is minimized?
Key Words

- **Reshaping** is about turning a matrix into a differently sized matrix or into a tensor or into a vector.

- The **Kronecker product** is an operation between two matrices that produces a highly structured block matrix.

- A **Rank-1 tensor** can be reshaped into a multiple Kronecker product of vectors.

- A **tensor decomposition** expresses a given tensor in terms of simpler tensors.

- The **alternating least squares** approach to multilinear LS is based on solving a sequence of linear LS problems, each obtained by freezing all but a subset of the unknowns.
What is the Course About?

The Lectures...

- Lecture 1. Introduction to Tensor Computations
- Lecture 2. Tensor Unfoldings
- Lecture 3. Transpositions, Kronecker Products, Contractions
- Lecture 4. Tensor-Related Singular Value Decompositions
- Lecture 5. The CP Representation and Rank
- Lecture 6. The Tucker Representation
- Lecture 7. Other Decompositions and Nearness Problems
- Lecture 8. Multilinear Rayleigh Quotients
- Lecture 9. The Curse of Dimensionality
- Lecture 10. Special Topics
# What is the Course About?

## The Next Big Thing

<table>
<thead>
<tr>
<th>Level</th>
<th>Time</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Scalar-Level Thinking</strong></td>
<td>1960’s</td>
<td>The factorization paradigm: $LU$, $LDL^T$, $QR$, $U\Sigma V^T$, etc.</td>
</tr>
<tr>
<td></td>
<td>1980’s</td>
<td>Cache utilization, parallel computing, LAPACK, etc.</td>
</tr>
<tr>
<td><strong>Matrix-Level Thinking</strong></td>
<td>2000’s</td>
<td>New applications, factorizations, data structures, nonlinear analysis, optimization strategies, etc.</td>
</tr>
<tr>
<td><strong>Block Matrix-Level Thinking</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Tensor-Level Thinking</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Lecture 1. Introduction to Tensor Computations
The Curse of Dimensionality

In Matrix Computations, to say that $A \in \mathbb{R}^{n_1 \times n_2}$ is “big” is to say that both $n_1$ and $n_2$ are big.

In Tensor Computations, to say that $A \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ is “big” is to say that $n_1 n_2 \cdots n_d$ is big and this need not require big $n_k$. E.g. $n_1 = n_2 = \cdots = n_{1000} = 2$.

Algorithms that scale with $d$ will induce a transition...

Matrix-Based Scientific Computation

\[ \downarrow \]

Tensor-Based Scientific Computation
The “Geometry” of the Tensor Research Community

Nonlinear Optimization

Applications

Nonlinear Analysis

Decompositions

Statistics

Multilinear Algebra

Programming Languages

Matrix Computations

Software Libraries

Transition to Computational Multilinear Algebra