Data Analysis in Fluid Mechanics

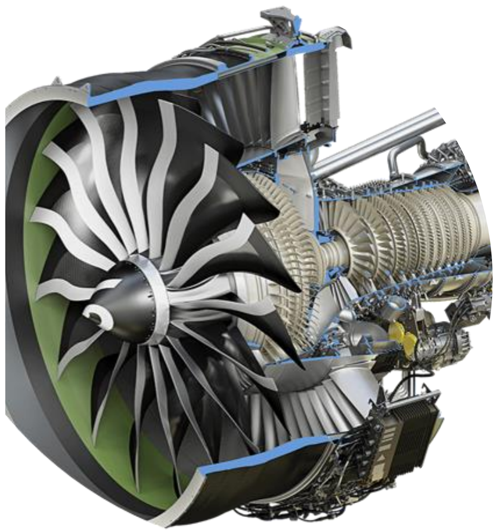
Project – 20%

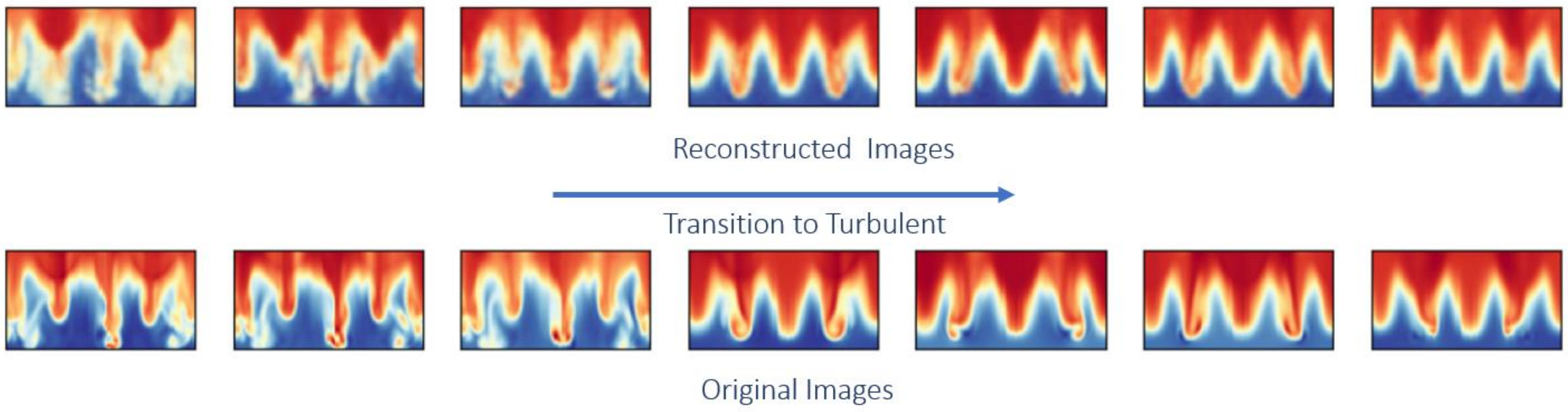
12th Feb 2024

ML in FM – An Introduction

- Tremendous data in fluids
- What to do with the data ?
- Improve performance / efficiency of systems

- Gas Turbines
- Automotive
- Electronic Cooling
- Heavy Industry
- Space
- Fundamental flows





Proper Orthogonal Decomposition

- The proper orthogonal decomposition is a numerical metho that enables a reduction in the complexity of computer intensive simulations such as computational fluid dynamics.
- **Proper Orthogonal Decomposition (POD)**, also known as **Principal Component Analysis (PCA)** in statistics and **Singular Value Decomposition (SVD)** in linear algebra, is a numerical method used to reduce the complexity of computer-intensive simulations. It finds applications in fields such as **computational fluid dynamics** and **structural analysis**.
- Why POD?

- To understand pertinent features in a dataset.

- Store, represent and efficiently manipulate data.

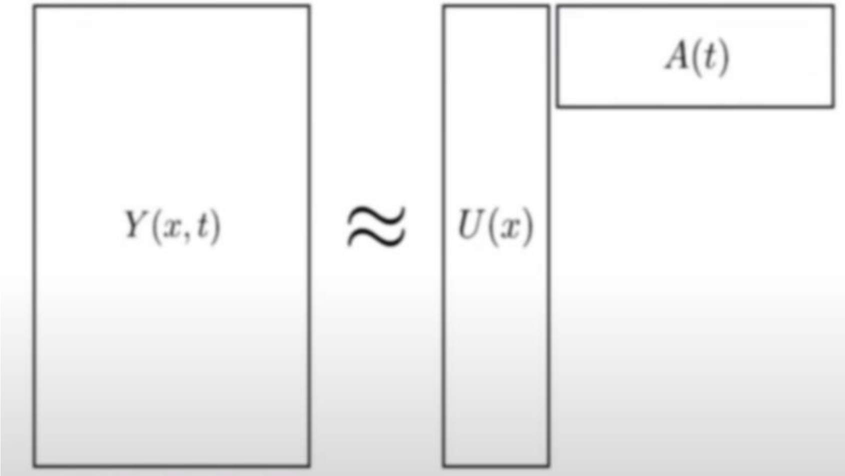
- Build a reduced-complexity model for the dynamics we are interested in.

POD

- **Purpose**: POD aims to replace complex models (such as the Navier–Stokes equations in fluid dynamics) with simpler ones based on simulation data
- **Decomposition**: It decomposes a physical field (e.g., pressure, temperature, stress, or deformation) into a set of deterministic spatial functions (called **modes** or **eigenfunctions**) modulated by random time coefficients.
- **Snapshot Method**: First, the vector field is sampled over time, creating snapshots. These snapshots are averaged over space dimensions and correlated along time samples.
- **Covariance Matrix**: The covariance matrix is computed from the snapshots.
- **Eigenvalues and Eigenvectors**: The eigenvalues and eigenvectors of the covariance matrix are computed and ordered from largest to smallest.
- **Representation**: The modes capture the dominant features of the field, allowing for a lower-dimensional representation.

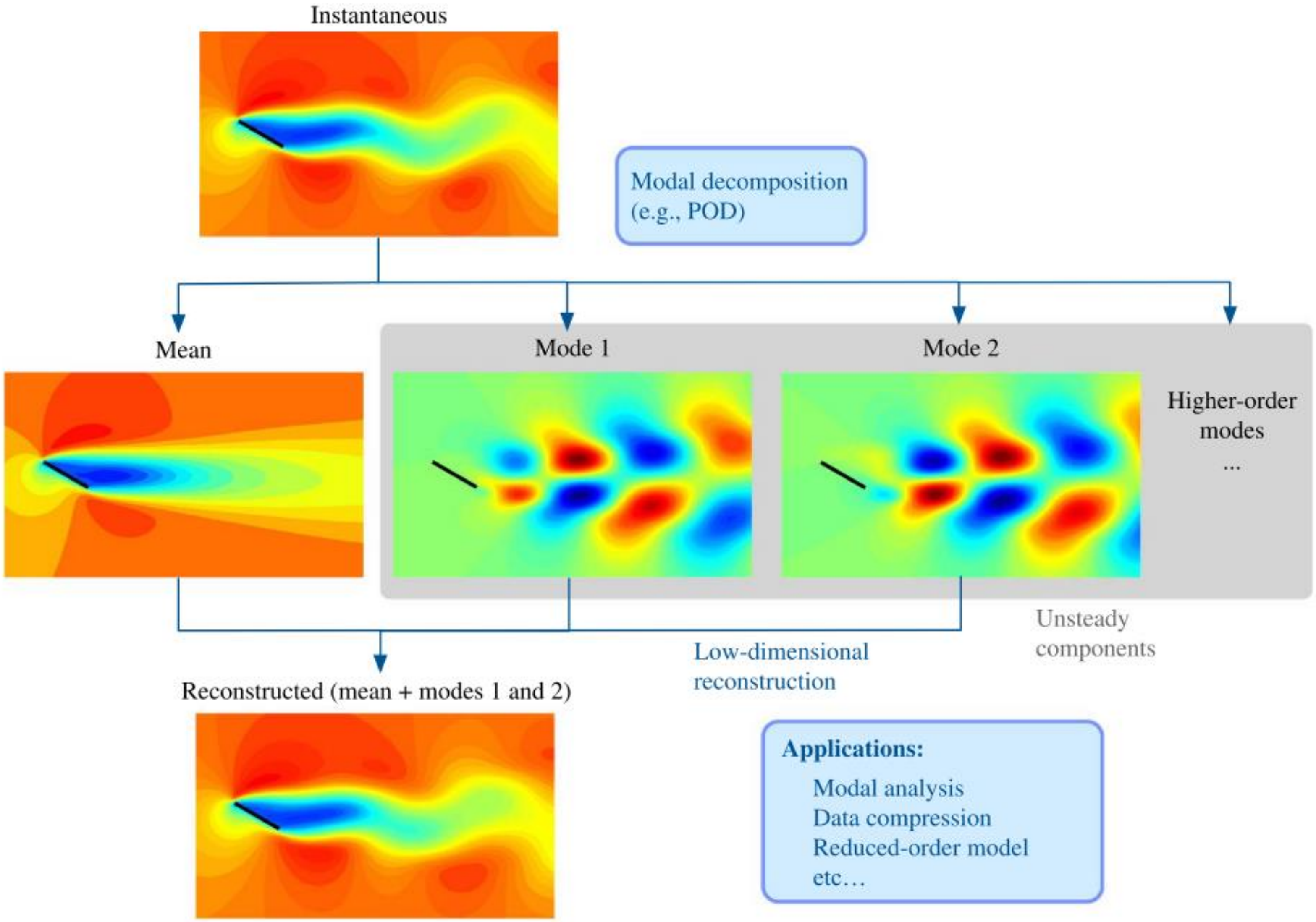
POD

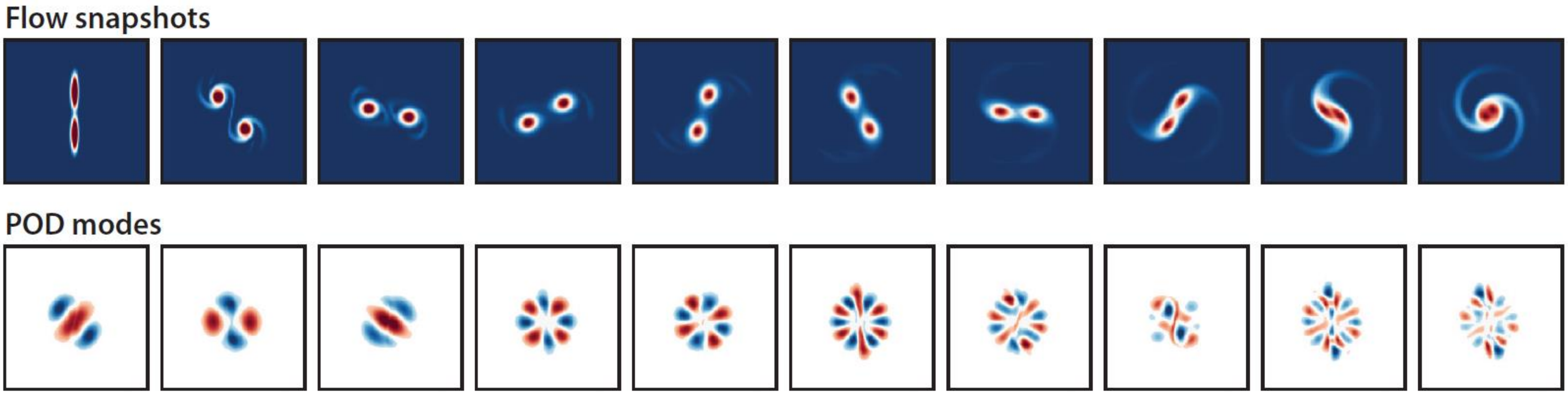
- Break up a function of space and time into individual spatial and temporal components.
- Arrange data into the respective matrix.



$$y(x,t) = \sum_{j=1}^{m} u_j(x) a_j(t)$$

An Example


Instantaneous
Modal decomposition
(e.g., POD)
Mean
Mode 1
Mode 2
Higher-order
modes
...
Unsteady
components
Low-dimensional
reconstruction
Reconstructed (mean + modes 1 and 2)
Applications:
Modal analysis
Data compression
Reduced-order model
etc…


Flow snapshots
POD modes

Eigenvalue Decomposition

$A \in \mathbb{C}^{n\times n}$, a vector $\boldsymbol{v} \in \mathbb{C}^n$ and a scalar $\lambda \in \mathbb{C}$

are called eigen vector and eigenvalue if A if they satisfy

$$A\boldsymbol{v} = \lambda\boldsymbol{v}$$

- The eigenvalues and eigenvectors of a matrix (linear operator) capture the directions in which vectors can grow or shrink.

If A has n linearly independent eigenvectors $\boldsymbol{v}_j$ with corresponding eigenvalues λ_j ($j = 1, \ldots, n$), then we have

$$AV = V\Lambda \tag{2}$$

where $V = [\boldsymbol{v}_1 \ \ \boldsymbol{v}_2 \ \ \ldots \ \ \boldsymbol{v}_n] \in \mathbb{C}^{n\times n}$ and $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n) \in \mathbb{C}^{n\times n}$. Postmultiplying V^{-1} to the preceding equation, we have

$$A = V\Lambda V^{-1} \tag{3}$$

Singular Value Decomposition

The SVD is a factorization of a $m \times n$ matrix into

$$\boldsymbol{A} = \boldsymbol{U}\,\boldsymbol{\Sigma}\,\boldsymbol{V}^{\boldsymbol{T}}$$

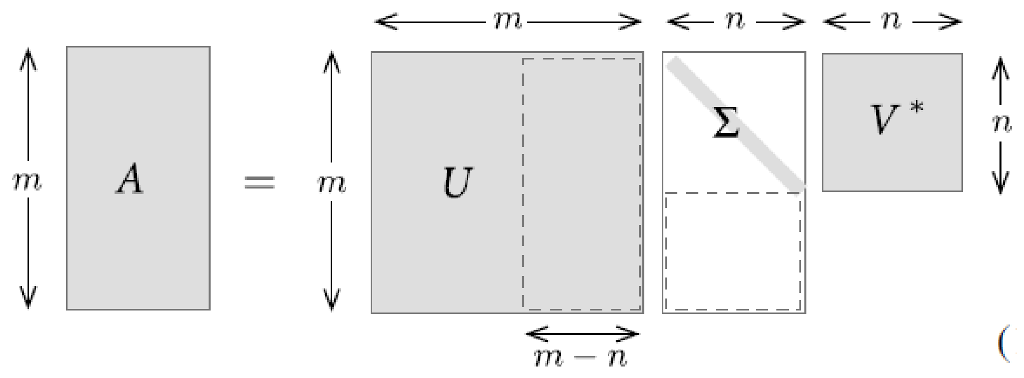
where $\boldsymbol{U}$ is a $m \times m$ orthogonal matrix, $\boldsymbol{V}^{\boldsymbol{T}}$ is a $n \times n$ orthogonal matrix and $\boldsymbol{\Sigma}$ is a $m \times n$ diagonal matrix.

For a square matrix ($\boldsymbol{m} = \boldsymbol{n}$):

$$\sigma_1 \geq \sigma_2 \geq \sigma_3 \ldots$$

$$\boldsymbol{A} = \begin{pmatrix} \vdots & \cdots & \vdots \\ \boldsymbol{u}_1 & \cdots & \boldsymbol{u}_n \\ \vdots & \cdots & \vdots \end{pmatrix} \begin{pmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \end{pmatrix} \begin{pmatrix} \cdots & \mathbf{v}_1^T & \cdots \\ \vdots & \vdots & \vdots \\ \cdots & \mathbf{v}_n^T & \cdots \end{pmatrix}$$

$$\boldsymbol{A} = \begin{pmatrix} \vdots & \cdots & \vdots \\ \boldsymbol{u}_1 & \cdots & \boldsymbol{u}_n \\ \vdots & \cdots & \vdots \end{pmatrix} \begin{pmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \end{pmatrix} \begin{pmatrix} \vdots & \cdots & \vdots \\ \boldsymbol{v}_1 & \cdots & \boldsymbol{v}_n \\ \vdots & \cdots & \vdots \end{pmatrix}^{\boldsymbol{T}}$$



Hint: look at Reduced SVD

POD Formulation

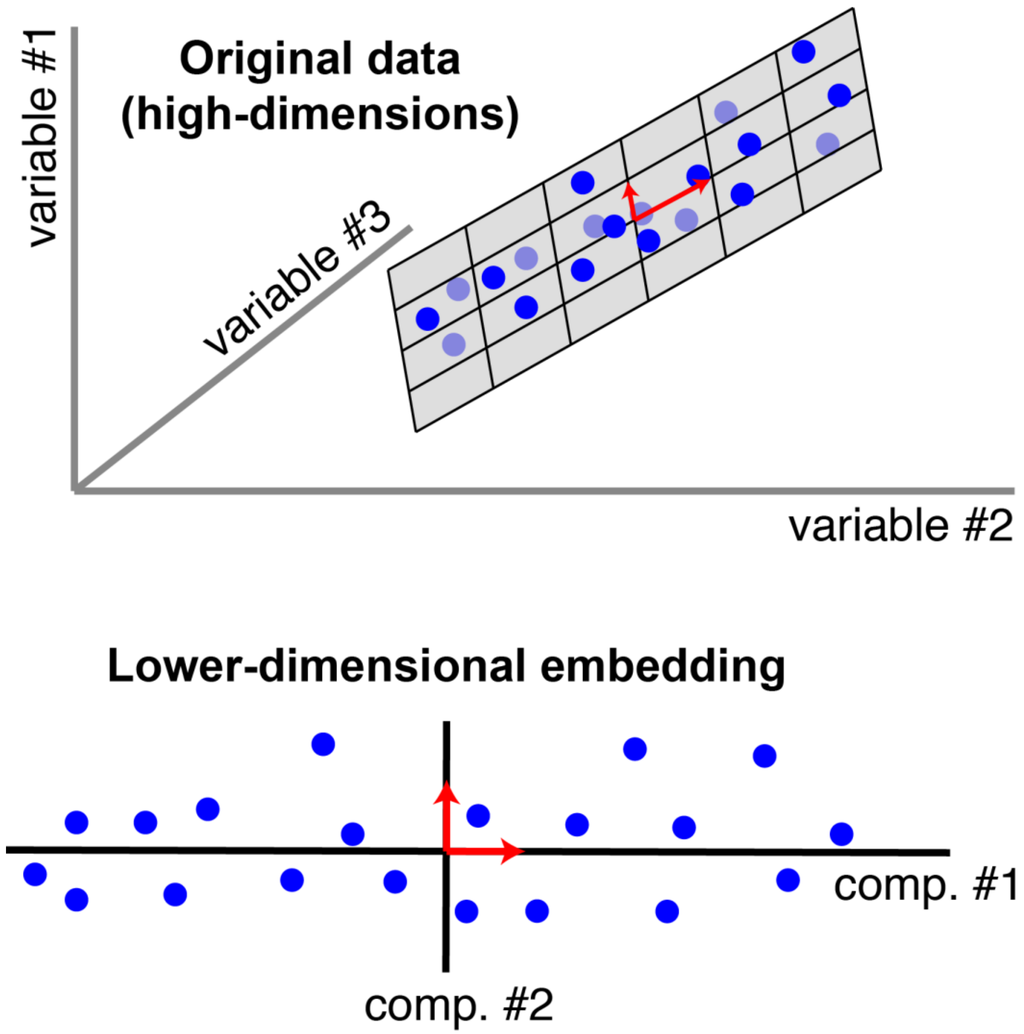
- SVD can be used OR
- The method of PCA – wherein an eigenvalue decomposition is performed for the covariance matrix R.

$$R\boldsymbol{\phi}_j = \lambda_j \boldsymbol{\phi}_j, \qquad \boldsymbol{\phi}_j \in \mathbb{R}^n, \qquad \lambda_1 \geq \ldots \geq \lambda_n \geq 0$$

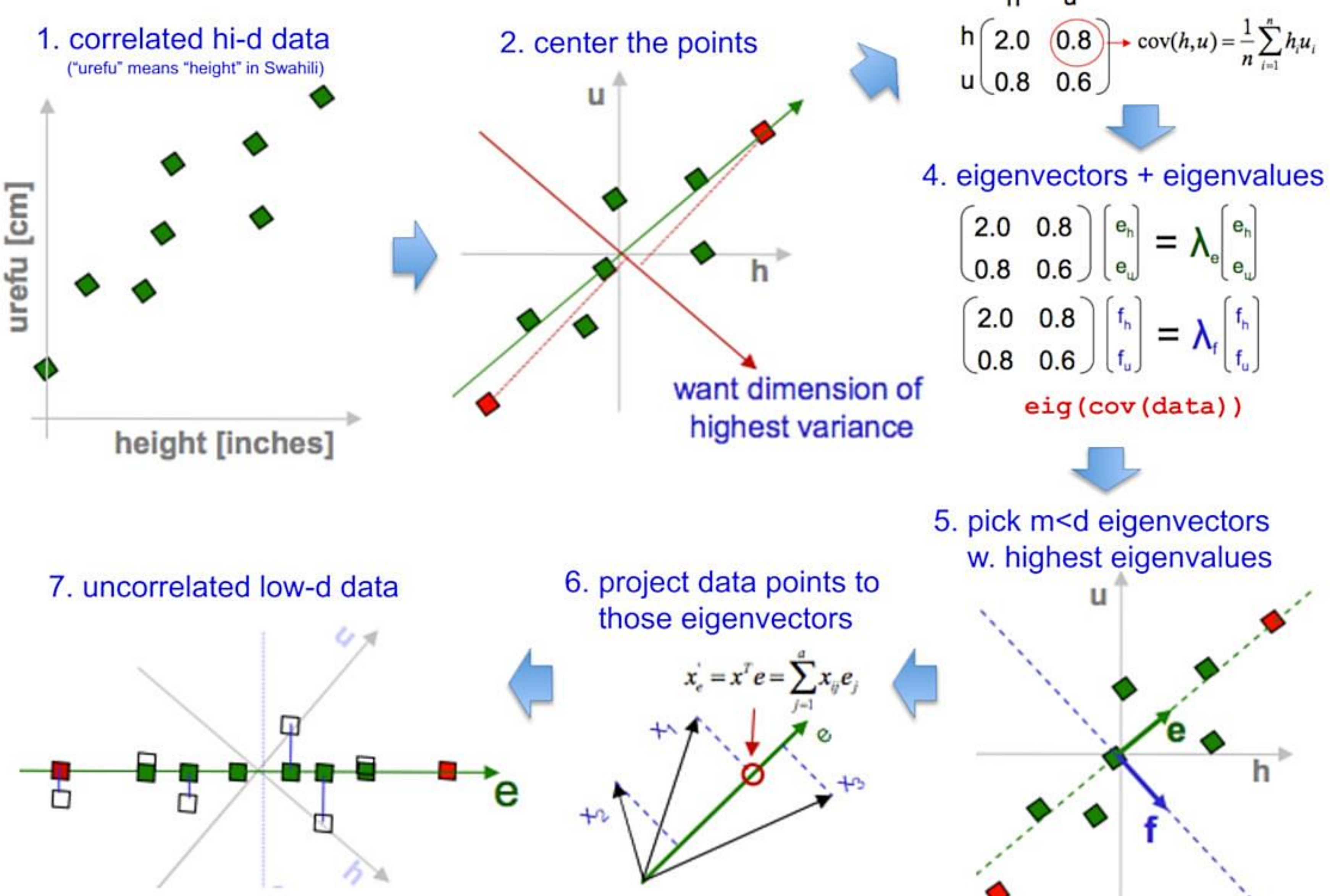
$$R = \sum_{i=1}^{m} \boldsymbol{x}(t_i)\boldsymbol{x}^T(t_i) = XX^T \in \mathbb{R}^{n\times n}$$

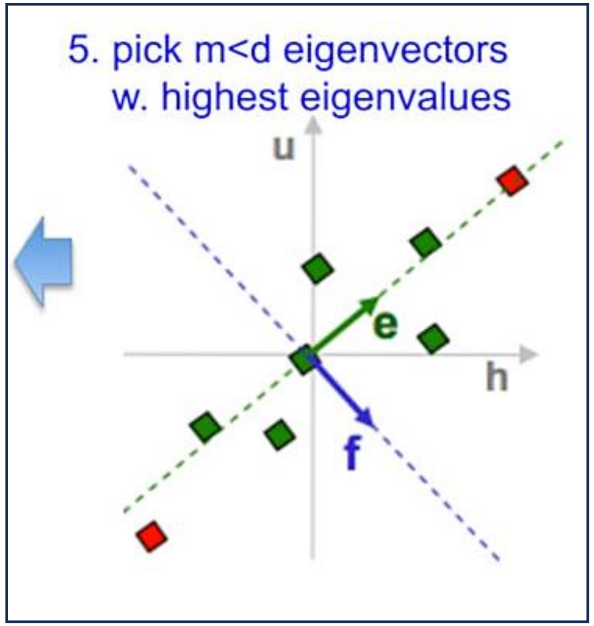
$$\operatorname{cov}(x_1, x_2, \ldots x_n) = \begin{bmatrix} \operatorname{var}(x_1) & \ldots & \operatorname{cov}(x_n, x_1) \\ \vdots & \ddots & \vdots \\ \operatorname{cov}(x_1, x_n) & \ldots & \operatorname{var}(x_n) \end{bmatrix}$$

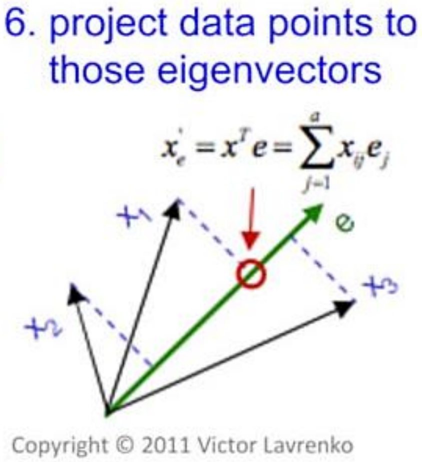
Principal Component Analysis

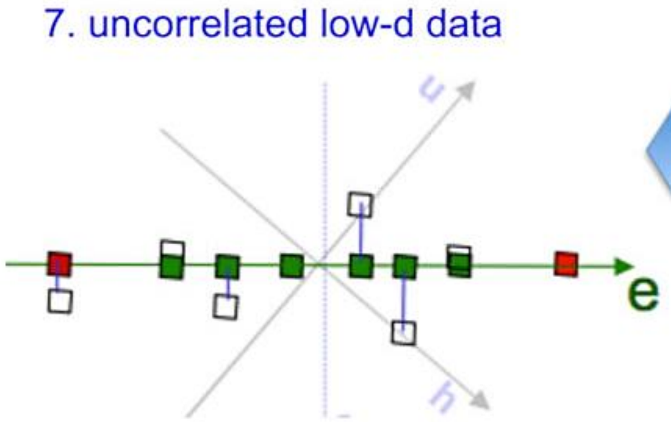


PCA in a nutshell









m eigenvectors with highest corresponding eigenvalues are the modes arranged in the ascending order

How does POD by PCA-covariance matrix method relate to the SVD Method?

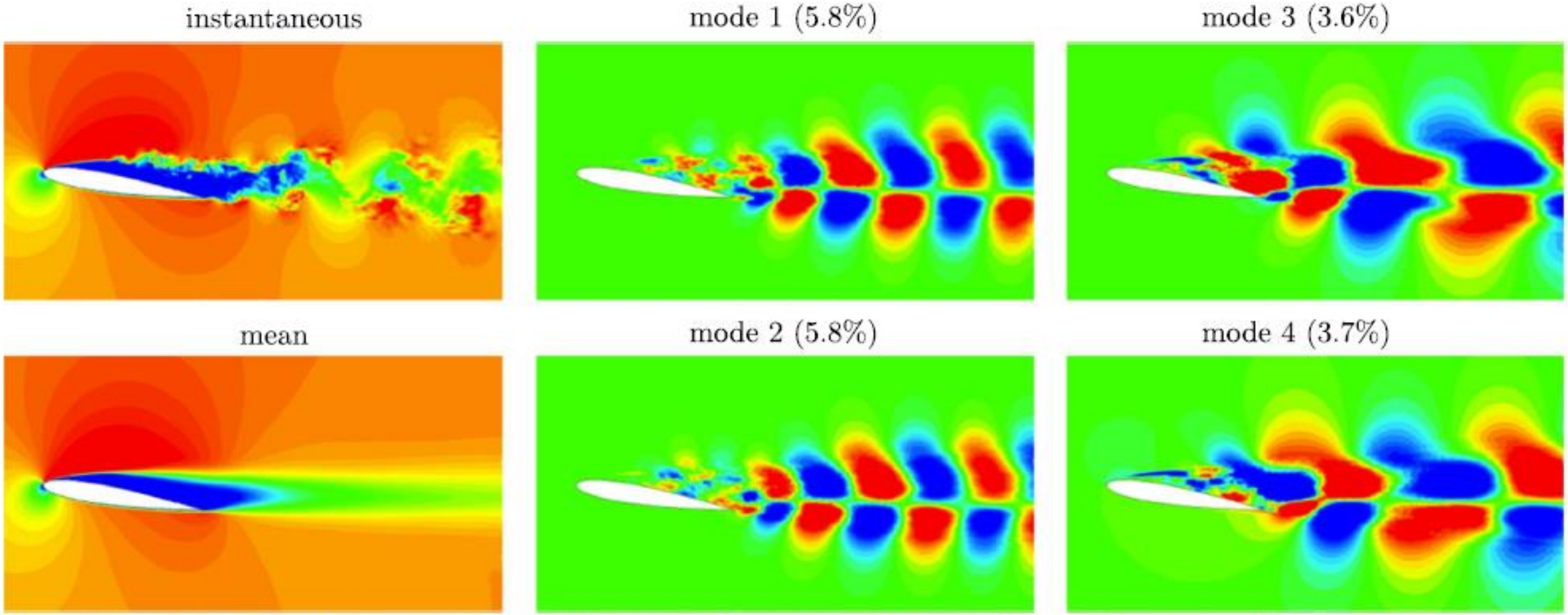
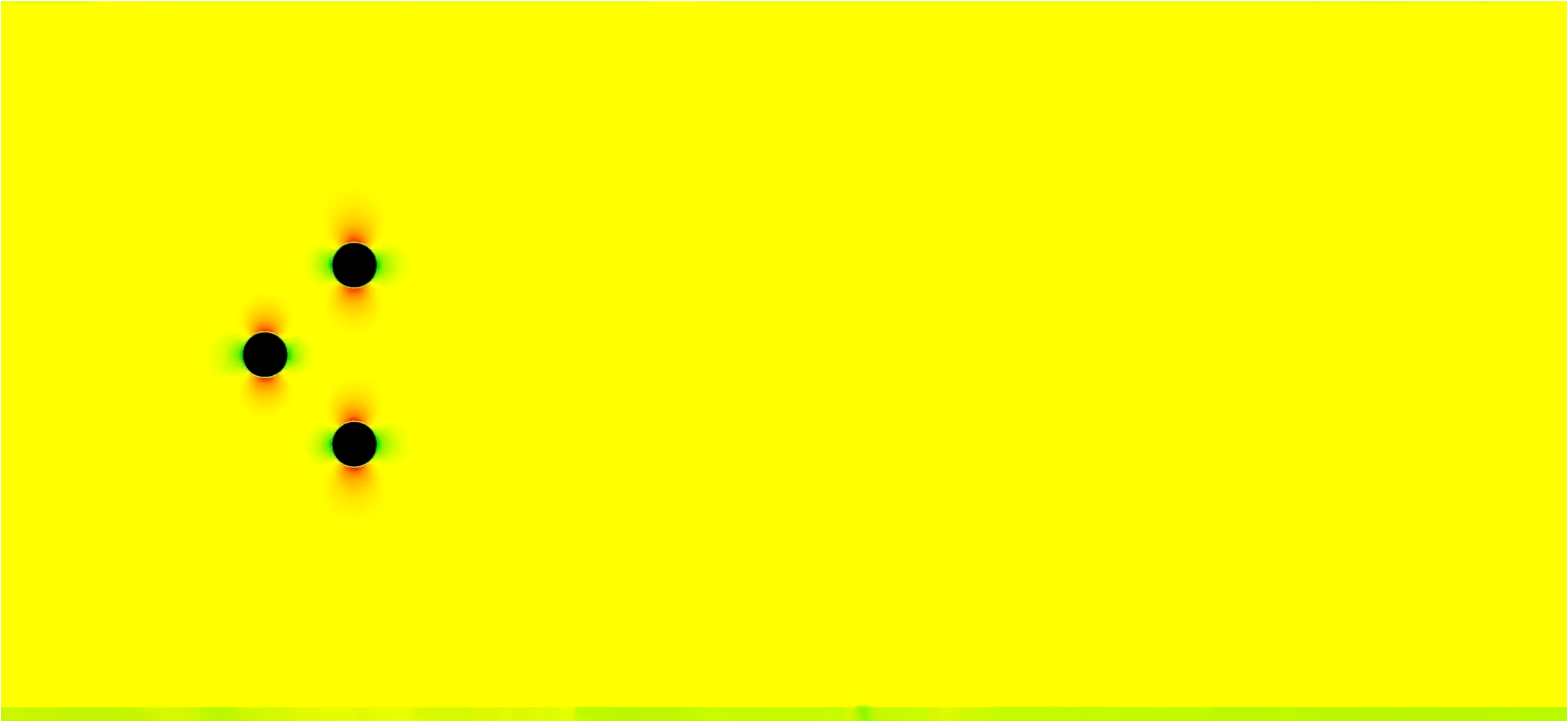


Fig. 6 POD analysis of turbulent flow over a NACA0012 airfoil at $Re = 23{,}000$ and $\alpha = 9$ deg. Shown are the instantaneous and time-averaged streamwise velocity fields and the associated four most dominant POD modes [73,74]. Reprinted with permission from Springer.

The Dataset … Video



Evaluation Guidelines

- 100 mark report (20% of FM course weightage)
- Due Date: 12th April 2024.
- 8% of the FM course weightage – presentation and viva post submission date.
- Groups of 2 only.
- Please use Python.
- Don’t ask ChatGPT – it will give you wrong answers! ☺
- Report submission will be through a plagiarism checking tool.

Report Marks

- Review of Machine Learning in Fluids (max 2 pages) – **15 marks**
- Any New Ideas (max. 2 pages) – **10 marks**
- Proper Orthogonal Decomposition – **30 marks**
- Noise – **25 marks**
- Super-Resolving – **20 marks**; 5 bonus marks for an exceptional effort.
- **3** bonus marks for writing the report in **LATEX**.

Reference Material

- Lots of it out there … please search
- Videos of Prof. Steve Brunton, others
- Search for research papers on Google Scholar.
- Ideation – Hint: look for recent trends in the ML domain and can you apply them to fluid mechanics.

Reduced SVD

What happens when $\boldsymbol{A}$ is not a square matrix?

1) $\boldsymbol{m > n}$

$$\boldsymbol{A} = \boldsymbol{U}\,\boldsymbol{\Sigma}\,\boldsymbol{V}^{\boldsymbol{T}} = \underbrace{\begin{pmatrix} \vdots & \cdots & \vdots & \cdots & \vdots \\ \boldsymbol{u}_1 & \cdots & \boldsymbol{u}_n & \cdots & \boldsymbol{u}_m \\ \vdots & \cdots & \vdots & \cdots & \vdots \end{pmatrix}}_{m\times m} \underbrace{\begin{pmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \\ & & 0 \\ & & \vdots \\ & & 0 \end{pmatrix}}_{m\times n} \underbrace{\begin{pmatrix} \cdots & \mathbf{v}_1^T & \cdots \\ \vdots & \vdots & \vdots \\ \cdots & \mathbf{v}_n^T & \cdots \end{pmatrix}}_{n\times n}$$

We can instead re-write the above as:

$$\boldsymbol{A} = \boldsymbol{U}_{\boldsymbol{R}}\,\boldsymbol{\Sigma}_{\boldsymbol{R}}\boldsymbol{V}^{\boldsymbol{T}}$$

Where $\boldsymbol{U}_{\boldsymbol{R}}$ is a $m\times n$ matrix and $\boldsymbol{\Sigma}_{\boldsymbol{R}}$ is a $n\times n$ matrix

Let's take a look at the product $\boldsymbol{\Sigma}^T\boldsymbol{\Sigma}$, where $\boldsymbol{\Sigma}$ has the singular values of a $\boldsymbol{A}$, a $m\times n$ matrix.

$$\boldsymbol{\Sigma}^T\boldsymbol{\Sigma} = \underset{n\times m}{\begin{pmatrix} \sigma_1 & & & 0 & & \\ & \ddots & & & \ddots & \\ & & \sigma_n & & & 0 \end{pmatrix}} \underset{m\times n}{\begin{pmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \\ & & 0 \\ & & \vdots \\ & & 0 \end{pmatrix}} = \underset{n\times n}{\boxed{\begin{pmatrix} \sigma_1{}^2 & & \\ & \ddots & \\ & & \sigma_n{}^2 \end{pmatrix}}}$$

$m > n$

$$\boldsymbol{\Sigma}^T\boldsymbol{\Sigma} = \underset{n\times m}{\begin{pmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_m \\ & & 0 \\ & & \vdots \\ & & 0 \end{pmatrix}} \underset{m\times n}{\begin{pmatrix} \sigma_1 & & & 0 & & \\ & \ddots & & & \ddots & \\ & & \sigma_m & & & 0 \end{pmatrix}} = \underset{n\times n}{\begin{pmatrix} \boxed{\begin{matrix} \sigma_1{}^2 & & \\ & \ddots & \\ & & \sigma_m{}^2 \end{matrix}} & \begin{matrix} 0 & & \\ & \ddots & \\ & & 0 \end{matrix} \\ \begin{matrix} 0 & & \\ & \ddots & \\ & & 0 \end{matrix} & \begin{matrix} 0 & & \\ & \ddots & \\ & & 0 \end{matrix} \end{pmatrix}}$$

$n > m$

POD/PCA

$$\mathbf{u}(\mathbf{x},t) \approx \bar{\mathbf{u}} + \sum_{k=1}^{r} \boldsymbol{\varphi}_k(\mathbf{x})\mathbf{a}_k(t)$$