Révision: Algèbre linéaire

(CIV8530 - Fiabilité des structures et systèmes)

Enseignant: James-A. Goulet

Département des génies civil, géologique et des mines
Polytechnique Montréal



Chapter 2 – Goulet (2020)
Probabilistic Machine Learning for Civil Engineers
MIT Press

Algèbre linéaire → Fondations



[Google images]

Plan du module #0a

Notation
Operations
Transformations

Organisation de la matière

Théorie probabilité
- 0 Révision algèbre & probabilités
- 1 Lois de probabilités
- 3 PDF multivariés

Fiabilité des structures & systèmes
- Introduction
- 2 Formulation fiabilité composantes
- 9 Formulation fiabilité systèmes

Estimation – p_f échantilonnage
- 4 Monte Carlo
- 12 Échantillonnage par importance

Estimation – p_f analytique
- 5 FOSM
- 6 FORM
- 8 SORM

Utilisation des résultats
- 7 Analyse de sensibilité
- 10 Incertitudes aléatoires & épistémiques
- 11 Données empiriques
- 13 Prise de décision
- 14 Métamodèles & modèles empiriques

Plan de la section

Notation

1.1 Scalaires

1.2 Vecteurs

1.3 Matrices

Scalaires

Scalaires $\rightarrow$ Nombre unique

$$
\begin{aligned}
x &\in \mathbb{R} && \equiv (-\infty, \infty) \\
&\in \mathbb{R}^{+} && \equiv (0, \infty) \\
&\in \mathbb{Z} && \equiv \{-\infty, \cdots, -1, 0, 1, 2, \cdots, \infty\} \\
&\in \ldots && (0, 1) \\
&\in \ldots && (0, 1]
\end{aligned}
$$

Vecteurs

Table 1-D contenant des nombres ou des scalaires

$$\mathbf{x} \equiv \underline{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

Si e.g., chaque $[\mathbf{x}]_i \in \mathbb{R}, \forall i = \{1 : n\} \rightarrow \mathbf{x} \in \mathbb{R}^n$

Matrices

Table 2-D contenant des nombres ou des scalaires

$$\mathbf{X} \equiv \underline{X} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix}$$

Si e.g., chaque $[\mathbf{X}]_{ij} \in \mathbb{R}, \forall i = \{1 : m\}, j = \{1 : n\} \rightarrow \mathbf{X} \in \mathbb{R}^{m \times n}$

Matrices carrées

Table 2-D contenant des nombres ou des scalaires

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nn} \end{bmatrix}$$

Si e.g., chaque $[\mathbf{X}]_{ij} \in \mathbb{R}, \forall i, j = \{1 : n\} \rightarrow \mathbf{X} \in \mathbb{R}^{n \times n}$

Matrices diagonales

Table 2-D contenant des nombres ou des scalaires sur sa diagonale principale et des 0 partout ailleurs.

$$\mathbf{Y} = \text{diag}(\mathbf{x}) = \begin{bmatrix} x_1 & 0 & \cdots & 0 \\ 0 & x_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & x_n \end{bmatrix}_{n \times n}$$

Matrice identité – $\mathbf{I}$

Une matrice identité est carrée ou les éléments de la diagonale principale sont 1 et 0 partout ailleurs.

$$\mathbf{I} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}_{n \times n}$$

$$\begin{aligned} \mathbf{I}\mathbf{x} &= \mathbf{x} \\ \mathbf{I}\mathbf{X} &= \mathbf{X} \end{aligned}$$

: Matrices

```python
import numpy as np
import scipy as sp

A = np.ones((3,2))                      #Matrice 3x2 remplie de 1
B = np.eye(5)                           #Matrice identite 5x5
C = np.array([[1, 2, 3], [4, 5, 6]])    #Matrice 2x3
D = C[1,2]                              #D=6
E = C[0,:]                              #E=[1 2 3]
F = np.diag(E)                          #F=Matrice diag. 3x3
```



Plan de la section

Operations

2.1 Transposition
2.2 Multiplication
2.3 Inversion de matrice

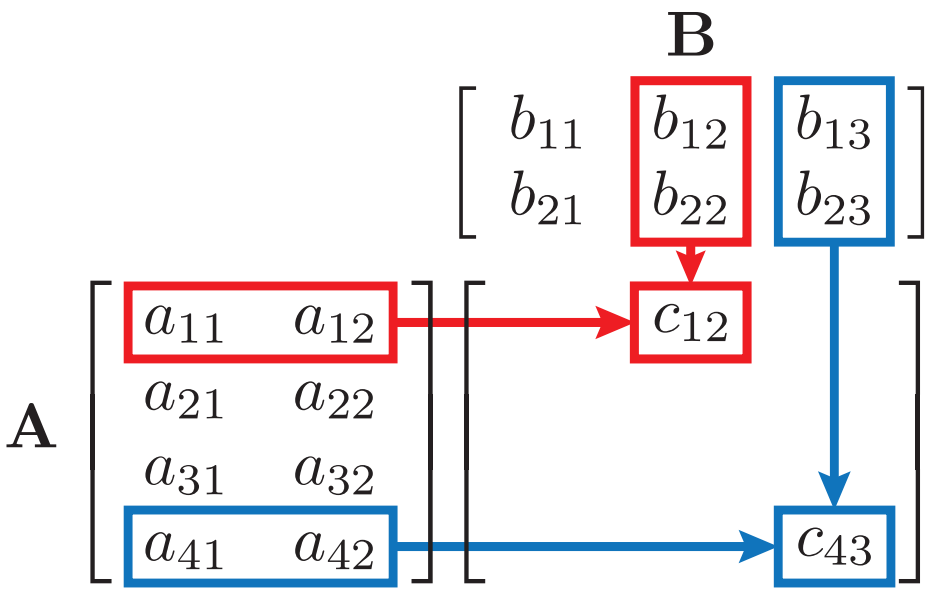
Transposition de matrices

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \end{bmatrix} \rightarrow \mathbf{X}^{\mathsf{T}} = \begin{bmatrix} x_{11} & x_{21} \\ x_{12} & x_{22} \\ x_{13} & x_{23} \end{bmatrix}$$

$$[\mathbf{X}^{\mathsf{T}}]_{ij} = [\mathbf{X}]_{ji}$$

Multiplication de matrices

$$\begin{aligned} \mathbf{C} &= \mathbf{AB} \\ &= \mathbf{A} \times \mathbf{B} \\ [\mathbf{C}]_{ij} &= \sum_k [\mathbf{A}]_{ik} \cdot [\mathbf{B}]_{kj} \\ \mathbf{C}_{m\times n} &= \mathbf{A}_{m\times k}\mathbf{B}_{k\times n} \end{aligned}$$



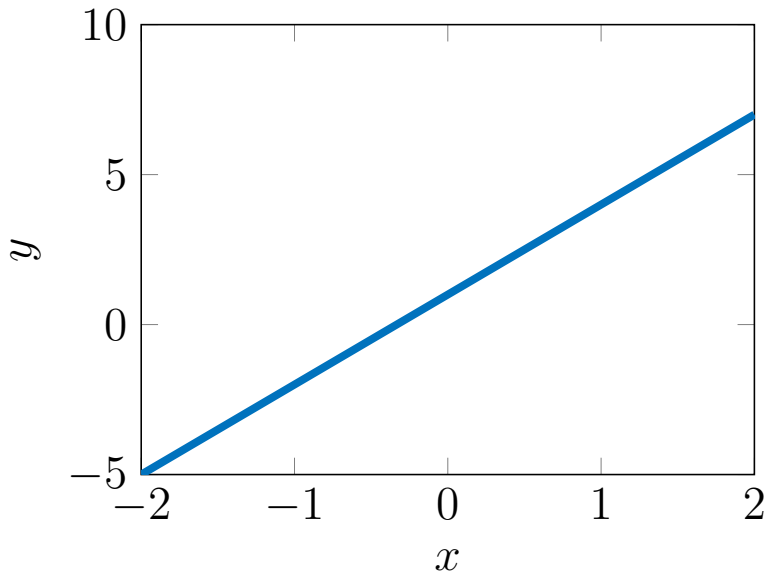
$$c_{12} = a_{11}b_{12} + a_{12}b_{22}$$

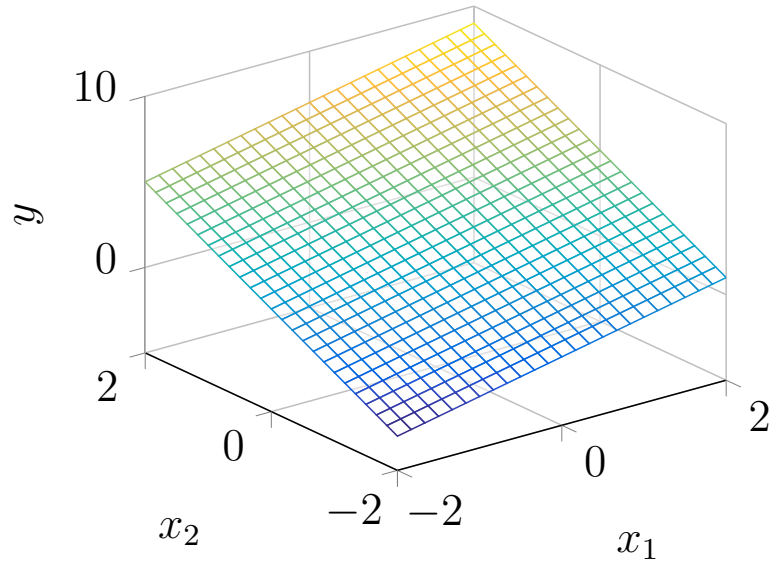
$$c_{43} = a_{41}b_{13} + a_{42}b_{23}$$

Règles de multiplication

$$\begin{aligned}
\mathbf{A}(\mathbf{B}+\mathbf{C}) &= \mathbf{AB}+\mathbf{AC} && \text{(Distributivité)} \\
\mathbf{A}(\mathbf{BC}) &= (\mathbf{AB})\mathbf{C} && \text{(Associativité)} \\
\mathbf{AB} &\neq \mathbf{BA} && \text{(Non commutativité)} \\
\mathbf{x}^{\mathsf{T}}\mathbf{y} &= \mathbf{x}\cdot\mathbf{y} && \text{(produit vectoriel)}
\end{aligned}$$

Systèmes linéaires





$$\begin{aligned} y &= 3x + 1 \\ &= ax + b \end{aligned}$$

$$a = 3, \; b = 1$$

$$\begin{aligned} y &= x_1 + 2x_2 + 3 \\ &= \mathbf{a}^\intercal \mathbf{x} + b \end{aligned}$$

$$\mathbf{a} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \; \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \; b = 3$$

Inversion de matrice

$$\mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$$

Pour être inversible, une matrice **doit être carrée** et **ne doit pas avoir de lignes ou colonnes linéairement dépendantes**.

Pour un système d'équations

$$\begin{aligned} \mathbf{A}\mathbf{x} &= \mathbf{b} \\ \mathbf{A}^{-1}\mathbf{A}\mathbf{x} &= \mathbf{A}^{-1}\mathbf{b} \\ \mathbf{I}\mathbf{x} &= \mathbf{A}^{-1}\mathbf{b} \\ \mathbf{x} &= \mathbf{A}^{-1}\mathbf{b} \end{aligned}$$

: Opérations

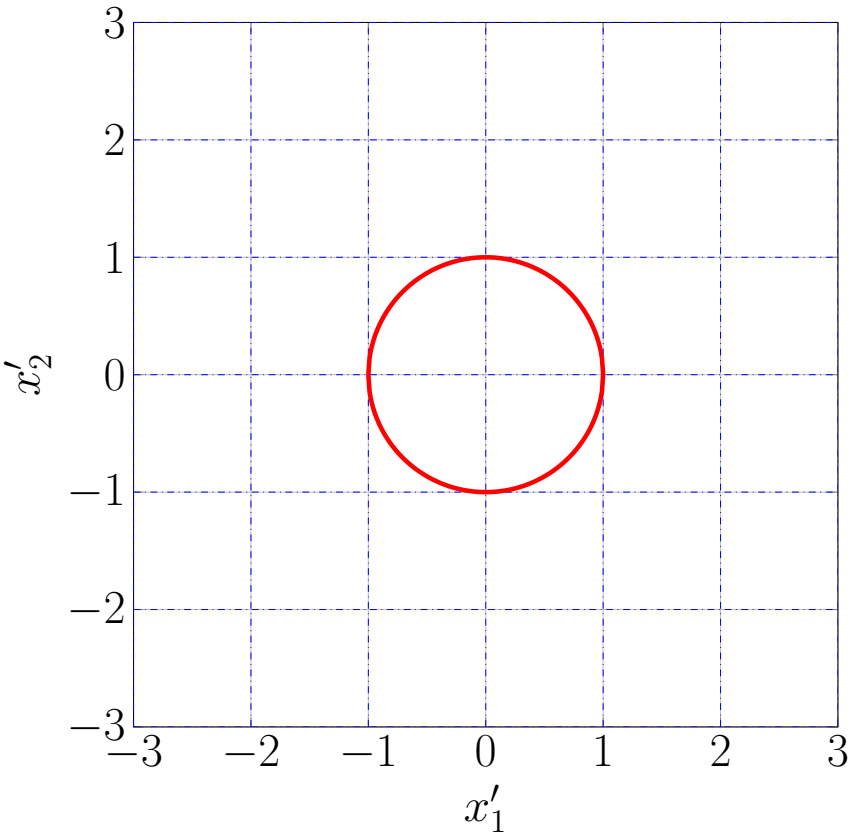
```python
import numpy as np

A = np.ones((3,2))                  #Matrice 3x2 remplie de 1
B = np.array([[1, 2, 3],[4, 5, 6]])#Matrice 2x3
C = B[0:1,0:1]
D = np.linalg.inv(C)                #D=inverse(C)
E = A @ B                           #Multiplication matricielle
F = A * B.transpose()               #Multiplication elements par
                                    elements
G = np.linalg.inv(F) @ B            #G=F^-1*B(1,:)
```

Transformations linéaires []

$$
\begin{aligned}
\mathbf{x}_1 &= [-1, -0.99, \cdots, 1]^\intercal \\
\mathbf{x}_2 : [\mathbf{x}_2]_i &= \sqrt{1 - [\mathbf{x}_1]_i^2} \\
\mathbf{X} &= [\mathbf{x}_1, \mathbf{x}_2] \\
\mathbf{X}' &= \mathbf{A}\mathbf{X}^\intercal
\end{aligned}
$$

$$
\mathbf{A} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}
$$

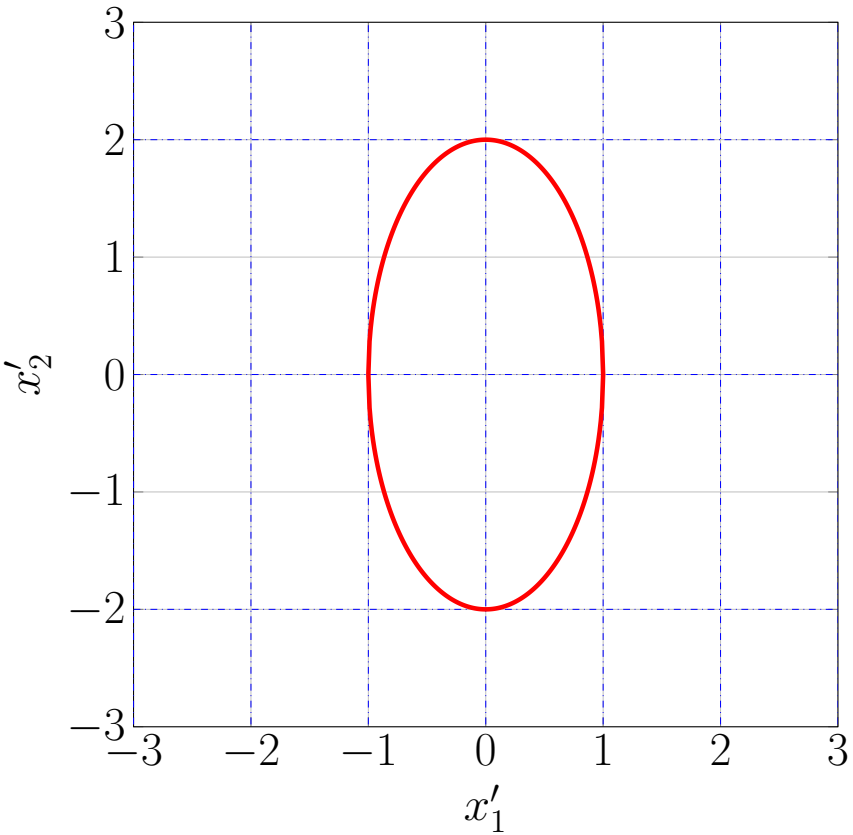


[CIVML/linear_transformation.py]

Transformations linéaires []

$$
\begin{aligned}
\mathbf{x}_1 &= [-1, -0.99, \cdots, 1]^\intercal \\
\mathbf{x}_2 : [\mathbf{x}_2]_i &= \sqrt{1 - [\mathbf{x}_1]_i^2} \\
\mathbf{X} &= [\mathbf{x}_1, \mathbf{x}_2] \\
\mathbf{X}' &= \mathbf{A}\mathbf{X}^\intercal
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{A} &= \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} \\
\det(\mathbf{A}) &= 2
\end{aligned}
$$

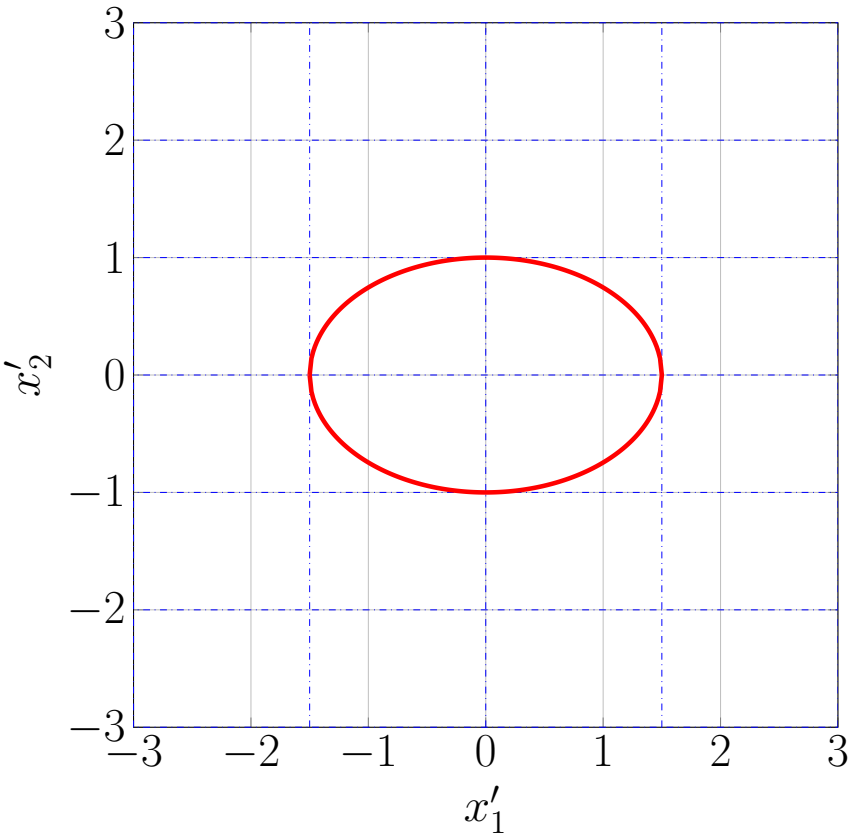


[CIVML/linear_transformation.py]

Transformations linéaires []

$$
\begin{aligned}
\mathbf{x}_1 &= [-1, -0.99, \cdots, 1]^\intercal \\
\mathbf{x}_2 : [\mathbf{x}_2]_i &= \sqrt{1 - [\mathbf{x}_1]_i^2} \\
\mathbf{X} &= [\mathbf{x}_1, \mathbf{x}_2] \\
\mathbf{X}' &= \mathbf{A}\mathbf{X}^\intercal
\end{aligned}
$$

$$
\mathbf{A} = \begin{bmatrix} 1.5 & 0 \\ 0 & 1 \end{bmatrix}
$$



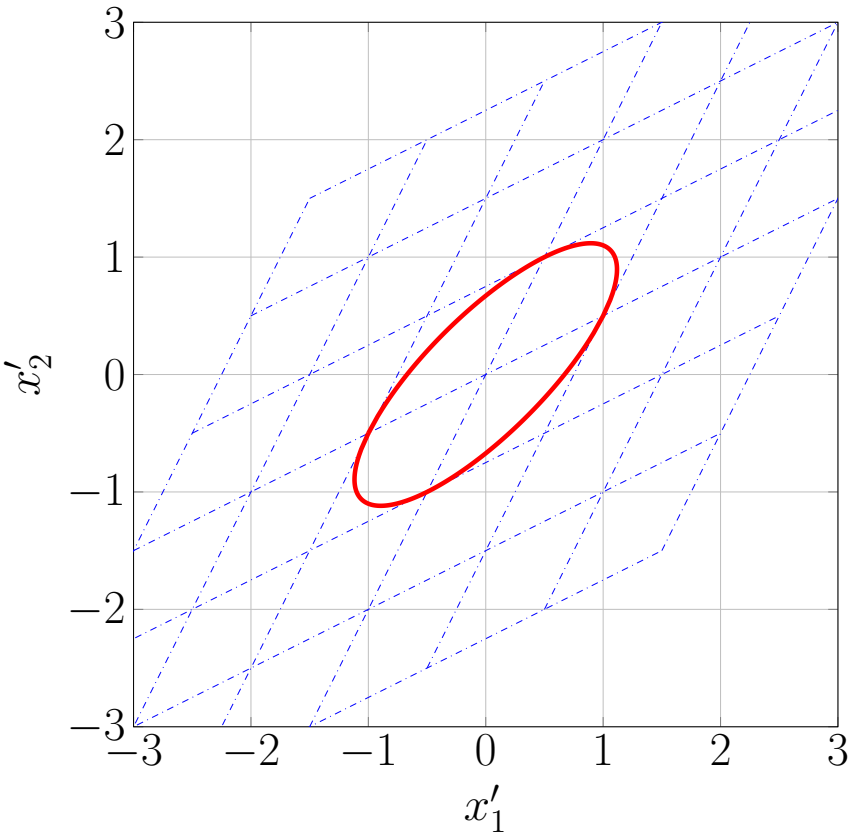
[CIVML/linear_transformation.py]

Transformations linéaires []

$$
\begin{aligned}
\mathbf{x}_1 &= [-1, -0.99, \cdots, 1]^\intercal \\
\mathbf{x}_2 : [\mathbf{x}_2]_i &= \sqrt{1 - [\mathbf{x}_1]_i^2} \\
\mathbf{X} &= [\mathbf{x}_1, \mathbf{x}_2] \\
\mathbf{X}' &= \mathbf{A}\mathbf{X}^\intercal
\end{aligned}
$$

$$
\mathbf{A} = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}
$$

$$
\det(\mathbf{A}) = 0.75
$$

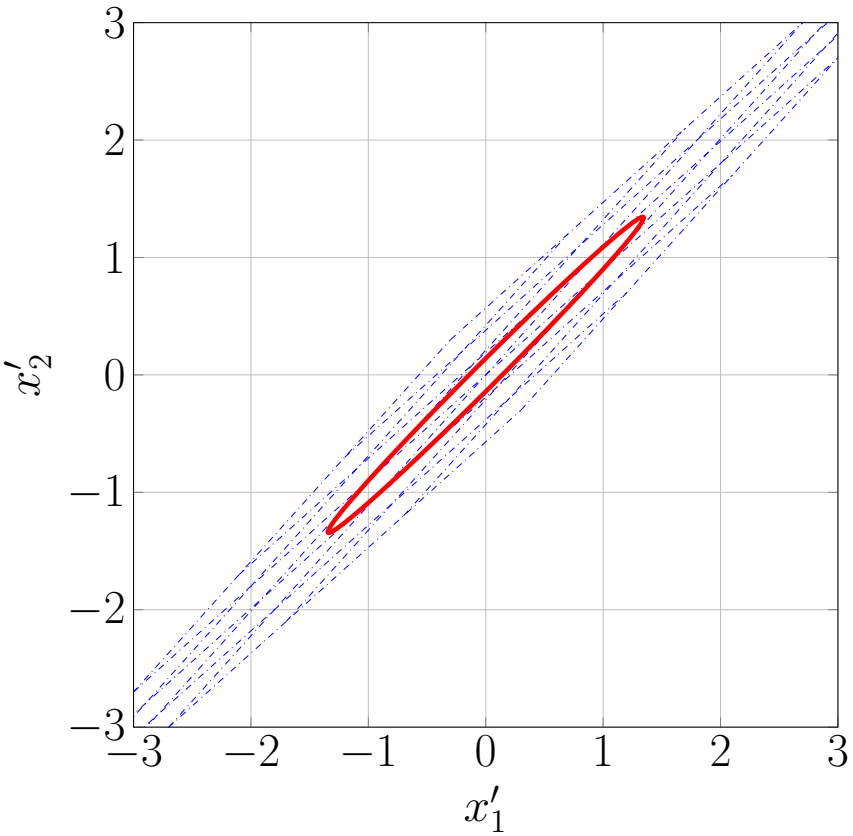


[CIVML/linear_transformation.py]

Transformations linéaires []

$$
\begin{aligned}
\mathbf{x}_1 &= [-1, -0.99, \cdots, 1]^\intercal \\
\mathbf{x}_2 : [\mathbf{x}_2]_i &= \sqrt{1 - [\mathbf{x}_1]_i^2} \\
\mathbf{X} &= [\mathbf{x}_1, \mathbf{x}_2] \\
\mathbf{X}' &= \mathbf{A}\mathbf{X}^\intercal
\end{aligned}
$$

$$
\mathbf{A} = \begin{bmatrix} 1 & 0.9 \\ 0.9 & 1 \end{bmatrix}
$$



[CIVML/linear_transformation.py]