

# Rapport - Double Descente - Emett Haddad

Emett Haddad, Encadrants: Nicolas Vayatis, Samuel Gruffaz

20/06/2024

## Contents

<b>1</b>	<b>Présentation du problème</b>	<b>1</b>
<b>2</b>	<b>Régimes et Double Descente</b>	<b>2</b>
<b>3</b>	<b>Modèles</b>	<b>3</b>
3.1	Modèle par MLP: (Multi Layer Perceptron) . . . . .	3
3.2	Modèle linéaire . . . . .	3
3.3	Modèle linéaire pénalisé (Ridge Regression) . . . . .	3
<b>4</b>	<b>Etat de l'art</b>	<b>4</b>
<b>5</b>	<b>Focus techniques</b>	<b>4</b>
5.1	Théorèmes de Belkin et al. . . . .	4
5.2	Théorème de Kuzborskij et al. . . . .	5
5.3	Théorème de Francis Bach . . . . .	5
<b>6</b>	<b>Résultats théoriques</b>	<b>6</b>
6.1	Régression linéaire . . . . .	6
6.2	Minimisation par descente de gradient . . . . .	9
6.2.1	Descente de gradient simple . . . . .	9
6.2.2	Descente de gradient à pas variable . . . . .	10
6.2.3	Analyse de la plus petite et plus grande valeur singulière . . . . .	12
6.2.4	Descente de gradient stochastique . . . . .	12
6.3	Expressions théorique sur les MLP . . . . .	12
<b>7</b>	<b>Résultats expérimentaux</b>	<b>13</b>
7.1	Régression polynomial . . . . .	13
7.2	Multilayer Perceptron (MLP) . . . . .	17
<b>8</b>	<b>Détails numériques</b>	<b>17</b>
<b>9</b>	<b>Appendices</b>	<b>18</b>
9.1	Pseudo-code . . . . .	19
9.1.1	Modèle linéaire et pénalisé . . . . .	19
9.1.2	Modèle MLP . . . . .	20

## 1 Présentation du problème

Un cours expliquant ce problème : [9]

### Définition 1. Fonction cible

On souhaite approximer une fonction  $y^*$  avec  $x$  et  $y$  des variables aléatoires.

$$y^* : x \in \mathcal{X} = \mathbb{R}^D \rightarrow \mathbb{E}_{(x,y) \sim \mathcal{P}}(y|x) \in \mathcal{Y} = \mathbb{R}$$

**Définition 2. Échantillons**

On se base sur un **échantillon d'apprentissage**  $\mathcal{D} := \{(x_n, y_n) \in \mathcal{X} \times \mathcal{Y}\}_{n=1}^N$  où  $y_n := y(x_n)$ . On modélise ici :  $y = y^* + \epsilon$  où  $\epsilon$  représente le bruit tel que  $\mathbb{E}(\epsilon|x) = 0$ , et  $\mathbb{V}(\epsilon) = \sigma^2$ .

**Définition 3. Estimateur**

- On veut trouver déterminer un **estimateur**  $\hat{y}_{\mathcal{D}} : \mathcal{X} \rightarrow \mathcal{Y}$  tel que  $\hat{y}_{\mathcal{D}} \approx y$  ( au sens du risque ci-dessous)
- $\hat{y}_{\mathcal{D}} \in \mathcal{H}$  un espace de fonctions

La **complexité** de  $\mathcal{H}$  est représentée ici par  $P$  le nombre de paramètres, bien que la meilleur notion actuelle de complexité est la complexité de Rademacher[9] qui est la capacité d'une classe de fonctions à s'adapter à du bruit.

**Définition 4. Erreur ponctuelle**

$\forall x_0 \in \mathcal{X}$

$$Err(x_0) := \mathbb{E}_{\mathcal{D}, \epsilon}((y - \hat{y})(x_0))^2 = \sigma^2 + \mathbb{V}_{\mathcal{D}}(\hat{y}(x_0)) + [\mathbb{E}_{\mathcal{D}}((y^* - \hat{y})(x_0))]^2 \quad (1)$$

Vrai risque et risque empirique:

$$\mathcal{R}(\hat{y}) = \mathbb{E}_{(x,y) \sim \mathcal{P}}((y - \hat{y}(x))^2), \quad \hat{\mathcal{R}}_{\mathcal{D}}(\hat{y}) = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}(x_i))^2 \quad (2)$$

## 2 Régimes et Double Descente

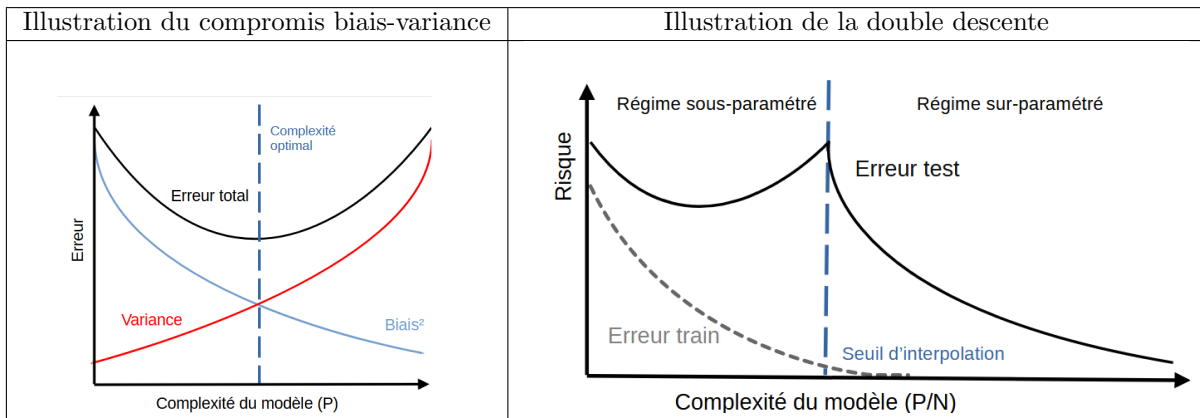
Notions de régimes:

- **Régime sous-paramétré:**  $P < N$  et **régime sur-paramétré:**  $P > N$
- **Seuil d'interpolation:**  $P = N$

**Définition 5. Notion de Double Descente:**

On dit qu'il y a **double descente** quand l'erreur global minimal dans le régime sur-paramétré est inférieure à celle dans le régime sous-paramétré.

En réalité, la double descente est ce même phénomène mais relativement à la complexité de la classe de fonctions[3].



### 3 Modèles

#### 3.1 Modèle par MLP: (Multi Layer Perceptron)

**Définition 6.** *Modèle MLP*

- $\hat{y}_{\mathcal{D}}(x) = W_L \circ \sigma \circ \dots \circ \sigma \circ W_1(x)$ ,  $W_l(x_l) = A_l x_l + b_l$  où  $A_l \in \mathcal{M}_{S_l, E_l}(\mathbb{R})$  et  $b_l \in \mathbb{R}^{S_l}$  tel que  $S_l = E_{l+1}$
- $\sigma(x) = \max(0, x)$  fonction ReLU.
- Par descente de gradient (stochastique) sur  $\hat{\mathcal{R}}_{\mathcal{D}}$  selon les paramètres  $A_l, b_l$ .

En pratique, on utilise ce type de modèle, mais leur résolution étant assez compliqué, on préfère étudier le modèle suivant plus simple.

#### 3.2 Modèle linéaire

**Définition 7.** *Modèle linéaire*

- $\mathcal{F} := \{f_i(x), i \in \mathbb{N}\}$  ensemble de features.
- $X = [x_1, \dots, x_N]^T \in \mathcal{M}_{N,D}(\mathbb{R})$ ,  $Y = [y_1, \dots, y_N]^T$
- $\Phi_P(x) = [f_1(x), \dots, f_P(x)]^T$  et  $Z = [\Phi_P(x_1), \dots, \Phi_P(x_N)]^T = [f_j(x_i)] \in \mathcal{M}_{N,P}(\mathbb{R})$
- $\hat{y}_{\mathcal{D}} = \Phi_P^T(x) \hat{\beta}$ , pour  $\hat{\beta} = Z^\dagger Y$
- Risque empirique  $\hat{\mathcal{R}}_{\mathcal{D}}(y) = \frac{1}{N} \|Y - Z\beta\|_2^2$
- Modèle simple  $f_i(x) = e_i(x)$  et  $X = Z$

**Définition 8.** *Optimisation par descente de gradient*

- Descente de gradient simple:

$$\hat{\beta}_{t+1} = \hat{\beta}_t - \alpha_t \nabla \hat{\mathcal{R}}_{\mathcal{D}}(\hat{y}_{\hat{\beta}_t})$$

- Descente de gradient stochastique:

$$\hat{\beta}_{t+1} = \hat{\beta}_t - \alpha_t J_t \nabla \hat{\mathcal{R}}_{\mathcal{D}_t}(\hat{y}_{\hat{\beta}_t})$$

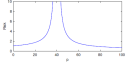
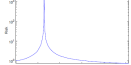
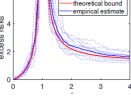
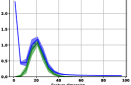
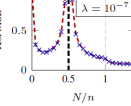
où  $J_t = \text{Diag}(\delta_{1,t} \dots \delta_{P,t})$  et  $\mathcal{D}_t \subset \mathcal{D}$ . Ce sont des variables aléatoires.  $J_t$  représente le fait qu'on ne calcule qu'une partie du gradient, et  $\mathcal{D}_t$  représente les batchs, on ne calcule le gradient que sur un sous ensemble de l'échantillon.

#### 3.3 Modèle linéaire pénalisé (Ridge Regression)

**Définition 9.** *Modèle linéaire pénalisé*

- $\hat{y}_{\mathcal{D},\lambda} = \Phi_P^T(x) \hat{\beta}_\lambda$ , pour  $\hat{\beta}_\lambda = (Z^T Z + \lambda I_P)^{-1} Z^T Y$
- Risque empirique pénalisé  $\hat{\mathcal{R}}_{\mathcal{D},\lambda}(\hat{y}_\beta) = \|Y - Z\beta\|_2^2 + \lambda \|\beta\|_2^2$  où  $\lambda > 0$

## 4 Etat de l'art

	Features/Modèle	Résultats	Méthodes	Graphe
Belkin et al. [2]	$f_p(x) = e_{i_p}(x)$ , coordonnées aléatoires	Expression du vrai risque, non-asymptotique	Inverse de Wishart	
Belkin et al. [2]	$f_p(\omega) = \omega^{i_p}$ sur le cercle complexe Racine de l'unité aléatoires pour l'échantillon $\mathcal{D}$	Équivalent du vrai risque, asymptotique	Transformée de Stieltjes	
Francis Bach [1]	$\hat{y}(x) = (S^T x)^T \hat{\beta}$ et $Z = XS$ où S matrice de vecteurs sub-Gaussien.	Équivalent du risque d'excès, asymptotique	Transformée de Stieltjes	
Kuzborskij et al. [6]	$f_p(x) = e_p(x)$ Descente de gradient et non pseudo-inverse	Majoration du vrai risque non-asymptotique	Inégalité de concentration	
Zhenyu et al. [7]	$f_p(x) = \cos((v_p, x))$ ou sin, pseudo-inverse	Limite MSE, asymptotique	Étude de la résolvante	

## 5 Focus techniques

### 5.1 Théorèmes de Belkin et al.

**Théorème 1.** *Belkin et al.- base canonique [2]*

Dans le cas où l'on sélectionne de manière uniforme  $P$  coordonnées parmi les  $D$  coordonnées de la base canonique comme features. On a ici  $X^T X$  matrice de Wishart.

- Si  $P < N - 1$ :  

$$\mathbb{E}((y - \hat{y})^2) = [(1 - \frac{P}{D})||\beta^*||^2 + \sigma^2](1 + \frac{P}{N-P-1})$$
- Si  $P > N + 1$ :  

$$\mathbb{E}((y - \hat{y})^2) = ||\beta^*||^2[1 - \frac{N}{D}(2 - \frac{D-N-1}{P-N-1})] + \sigma^2(1 + \frac{N}{P-N-1})$$
- Sinon:  

$$\mathbb{E}((y - \hat{y})^2) = +\infty$$

**Démonstration.** Dans les grandes lignes

On utilise les propriétés du pseudo-inverse (propriété de projecteur) et des manipulations sur la trace. On utilise ensuite l'espérance de l'inverse de Wishart.

## 5.2 Théorème de Kuzborskij et al.

**Théorème 2.** *Kuzborskij et al[6] en régime sur-paramétré*

$E_T := \mathcal{R}(\hat{y}_{\beta_T}) - \mathcal{R}(\hat{y}_{\beta_*})$  où  $\beta_T$  est le vecteur obtenu au bout de  $T$  itérations de descente de gradient de pas  $\alpha$ .

$$\mathcal{E}_T \leq \left[ \left( 1 - \frac{2\alpha}{N} (\sqrt{D} - \sqrt{N} - 1)_+^2 \right)^{2T} + \frac{1}{\sqrt{N}} \right] \|\beta^*\|^2 \quad (3)$$

## 5.3 Théorème de Francis Bach

**Théorème 3.** *Francis Bach[1]*

On pose  $df_i(\lambda) = \text{tr}(\Sigma^i(\Sigma + \lambda I)^{-i})$ ,  $y = x^T S \beta^* + \epsilon$ . On pose  $y = x^T S \beta^* + \epsilon$ ,  $df_1(K_P) \sim P$ ,  $df_1(K_N) \sim N$ . Alors:

- Si  $\delta < 1$ :

$$\mathbb{E}(R^{(var)}(\hat{\beta})) \sim \frac{\sigma^2 \delta}{1 - \delta}$$

$$R^{(biais)}(\hat{\beta}) \sim \frac{K_P}{1 - \delta} \|\beta^*\|_{\Sigma(\Sigma + K_P I)^{-1}}^2$$

- Si  $\delta > 1$ :

$$\mathbb{E}(R^{(var)}(\hat{\beta})) \sim \frac{\sigma^2}{\delta - 1} + \frac{\sigma^2 df_2(K_N)}{N - df_2(K_N)}$$

$$R^{(biais)}(\hat{\beta}) \sim \frac{N K_N^2}{N - df_2(K_N)} \|\beta^*\|_{\Sigma(\Sigma + K_N I)^{-2}}^2 + \frac{K_N}{\delta - 1} \|\beta^*\|_{\Sigma(\Sigma + K_N I)^{-1}}^2$$

**Démonstration.** -

On utilise la transformée de Stiejes, et ses propriétés. On passe par la limite  $\lambda \rightarrow 0$  de la solution du problème pénalisé.

## 6 Résultats théoriques

### 6.1 Régression linéaire

On considère:  $\Phi_P(x) = [f_1(x), \dots, f_P(x)]^T$  où  $f_p : \mathbb{R}^D \rightarrow \mathbb{R}$  et  $\hat{y}(x) = \Phi_P^T(x)\hat{\beta}$ . On considère ici des fonctions de "moyenne" nulle i.e.  $\forall p \geq 1, \mathbb{E}_x(f_p(x)) = 0$

Avec  $(f, g)_X = \sum_{n=1}^N f(x_n)g(x_n)$ ,  $\{f, g\} = \frac{1}{2\pi} \int_0^{2\pi} f(e^{i\theta})\overline{g(e^{i\theta})}$  et  $G_P^x(z) = \sum_{p=1}^P f_p(x)z^p$

**Théorème 4.** *Expressions de l'estimateur*

$$\forall x \in \mathcal{X}, \hat{y}(x) = [f_1(x), \dots, f_P(x)][(f_i, f_j)_X]^\dagger \begin{bmatrix} (f_1, y)_X \\ \vdots \\ (f_P, y)_X \end{bmatrix} \quad (4)$$

$$\forall x \in \mathcal{X}, \hat{y}(x) = ([\{G_P^{x_i}, G_P^{x_j}\}]^\dagger [\{G_P^x, G_P^{x_j}\}])^T Y \quad (5)$$

**Démonstration.** On a de manière général,  $X^\dagger = (X^T X)^\dagger X^T = X^T (X X^T)^\dagger$ . On applique donc cela à l'égalité  $\hat{\beta} = Z^\dagger Y$ , puis à  $\hat{y}(x) = \Phi_P^T(x)\hat{\beta}$ . Enfin, on remarque :

$$\{G_P^{x_i}, G_P^{x_j}\} = \frac{1}{2\pi} \int_0^{2\pi} G_P^{x_i}(e^{i\theta})\overline{G_P^{x_j}(e^{i\theta})} = \frac{1}{2\pi} \int_0^{2\pi} \left(\sum_{p=1}^P f_p(x_i)e^{ip\theta}\right) \left(\sum_{p=1}^P f_p(x_j)e^{-ip\theta}\right) = \sum_{p=1}^P f_p(x_i)f_p(x_j)$$

**Théorème 5.** *Décroissance du paramètre  $\hat{\beta}$*

Dans le régime sur-paramétré,  $\|\hat{\beta}(P)\|_2$  est décroissante à partir du moment où le rang de la matrice  $Z$  devient maximale i.e  $\text{rg}(Z) = N$ .

**Démonstration.** En effet, en supposant  $\text{rg}(Z) = N$ , nous avons ce résultat par l'étude du Lagrangien [8], qu'il s'agit du minimum global pour la norme euclidienne dans l'espace affine  $\hat{\beta} + \text{Ker}(Z) = \{\beta, Y = Z(P)\beta\}$ :

$$\forall \beta \in \mathbb{R}^P, \forall \gamma \in \mathbb{R}^N \mathcal{L}_P(\beta, \gamma) = \|\beta\|_2^2 + \gamma^T (Y - Z(P)\beta) \quad (6)$$

En effet la solution de  $\nabla \mathcal{L}_P(\beta, \gamma) = 0$  est  $\hat{\beta}(P)$ . C'est donc l'unique solution de  $Y = Z(P)\beta$  de norme minimale.

Et on a  $\hat{\beta}'(P+1) := \begin{bmatrix} \hat{\beta}(P) \\ 0 \end{bmatrix}$  est aussi solution de  $Y = Z(P+1)\beta$  donc  $\|\hat{\beta}(P+1)\|_2 \leq \|\hat{\beta}'(P+1)\|_2 = \|\hat{\beta}(P)\|_2$ .

On remarque que ce phénomène ( $\text{rg}(Z) = N$ ) survient APCR dans le cas de la régression polynomial à  $D$  variables car il suffit de pouvoir extraire une matrice de Vandermonde (quitte à changer de base) par exemple, quelle que soit la dimension.

On note  $E$  paramètres pour l'espace global, et on note  $P$  paramètres et  $P^C$  les autres paramètres parmi ces  $E$  paramètres. On rappelle  $Z_P = \Phi_P^T(X)$ ,  $Z_{P^C} = \Phi_{P^C}^T(X)$ ,  $Z_E = [Z_P, Z_{P^C}] = \Phi_E^T(X)$

**Théorème 6.** *Expression de l'excès de risque renormalisé*

On pose  $\Phi_E(x) = \begin{bmatrix} \Phi_P(x) \\ \Phi_{P^C}(x) \end{bmatrix}$  et  $\mathcal{E}(\hat{y}, \beta^*) := \mathcal{R}(\hat{y}) - \mathcal{R}(y^*)$ , où l'on suppose  $y^*(x) = \Phi_E(x)^T \beta^*$  avec  $E \in \mathbb{N}$  features orthonormées. On note  $\mathbb{E}(\Phi_E(x)\Phi_E(x)^T) = \sigma_\Phi^2 I_E$ .

On a alors:

$$\sigma_\Phi^{-2} \mathcal{E}(\hat{y}_D, \beta^*) = \|\hat{\beta} - \beta^*\|^2 = \|\beta^*\|^2 + \|Z_P^\dagger Z_{P^C} \beta_{P^C}^*\|^2 - \text{tr}[(Z_P^\dagger Z_P) \beta_P^* \beta_P^{*T}] \quad (7)$$

**Démonstration.** On s'inspire de la preuve de Belkin et al [2]. On note de même  $\beta_P$  pour parler de  $P$  composante de  $\beta$ , de même pour les vecteurs colonnes des matrices. On note  $\beta_{PC}$  pour les autres composantes.

$\mathcal{R}(\hat{y}) = \mathbb{E}_P((y - \hat{y}(x))^2) = \mathcal{R}(y^*) + \mathbb{E}_P((y^*(x) - \hat{y}(x))^2)$  car  $\mathbb{E}(y(x) - y^*|x) = 0$  par hypothèse sur le bruit.

D'où on a  $\mathcal{R}(\hat{y}_D) - \mathcal{R}(y^*) = \text{tr}[\mathbb{E}_x(\Phi_E(x)\Phi_E(x)^T)(\hat{\beta} - \beta^*)(\hat{\beta} - \beta^*)^T] = \sigma_\Phi^2 \|\hat{\beta} - \beta^*\|^2$

On a  $Y = Z_E \beta^*$  et  $Z_E = [Z_P, Z_{PC}]$  D'où  $Y = Z_P \beta_P^* + Z_{PC} \beta_{PC}^*$ . On pose  $\Pi_{Z_P} := Z_P^\dagger Z_P$  le projecteur orthogonale sur  $\text{Im}(Z_P)$ . De plus on a  $\hat{\beta}_P = Z_P^\dagger Y$  et  $\hat{\beta}_{PC} = 0$ .

On a alors:  $\|\hat{\beta} - \beta^*\|^2 = \|\hat{\beta}_P - \beta_P^*\|^2 + \|\beta_{PC}^*\|^2 = \|(\Pi_{Z_P} - I_P)\beta_P^*\|^2 + \|Z_P^\dagger Z_{PC} \beta_{PC}^*\|^2 + \|\beta_{PC}^*\|^2$ , par les propriétés de projecteur.

De plus on a de même:  $\|(\Pi_{Z_P} - I_P)\beta_P^*\|^2 = \text{tr}[\beta_P^{*T}(\Pi_{Z_P} - I_P)^T(\Pi_{Z_P} - I_P)\beta_P^*] = \|\beta_P^*\|^2 - \text{tr}[\Pi_{Z_P} \beta_P^* \beta_P^{*T}]$ .

D'où  $\mathcal{E}(\hat{y}_D, \beta^*) = \sigma_\Phi^2 \cdot (\|\beta^*\|^2 + \|Z_P^\dagger Z_{PC} \beta_{PC}^*\|^2 - \text{tr}[\Pi_{Z_P} \beta_P^* \beta_P^{*T}])$ .

Remarque: Dans le cas d'une base orthonormée sur un cube  $I$ , avec  $x$  suivant une loi uniforme, on a  $\mathbb{E}_{x \sim \mathcal{U}(I)}(f_i(x)f_j(x)) = \sigma_\Phi^2 \delta_{i,j}$ . Ainsi on a bien:  $\mathbb{E}_x(\Phi_E(x)\Phi_E(x)^T) = \sigma_\Phi^2 I_E$ .

De plus on fixe  $f_0(x) := 1$  et  $\forall p \geq 1$ ,  $\mathbb{E}_x(f_p(x)) = \mathbb{E}_x(f_0(x)f_p(x)) = 0$ . D'où en enlevant la constante, on a  $\mathbb{E}(\Phi_P(x)) = 0$ . Ainsi, en posant  $z_i = \mathbb{E}(\Phi_P(x_i)) \in \mathbb{R}^P$  on a les  $z_i$  sont indépendants par lemme de coalition, et de moyenne nulle, et même variance, on est donc dans le contexte de "Marchenko–Pastur distribution".

**Corollaire 1.** Expression de l'excès de risque

On modélise ici  $\beta^*$  par une variable aléatoire tel que  $\mathbb{E}(\beta^* \beta^{*T}) = \sigma_{\beta^*}^2 I_E$ . On pose  $\mathcal{E}(\hat{y}) := \mathbb{E}_{\beta^*}(\mathcal{E}(\hat{y}, \beta^*))$ .

$$(\sigma_\Phi \sigma_{\beta^*})^{-2} \mathcal{E}(\hat{y}_D) = E - 2rg(Z_P) + \text{tr}[Z_E Z_E^T (Z_P Z_P^T)^\dagger] \quad (8)$$

**Démonstration.** On a  $\|Z_P^\dagger Z_{PC} \beta_{PC}^*\|^2 = \text{tr}[Z_{PC}^T (Z_P Z_P^T)^\dagger Z_{PC} \beta_{PC}^* \beta_{PC}^{*T}]$  d'où:

$$\sigma_\Phi^{-2} \mathcal{E}(\hat{y}_D, \beta^*) = \|\beta^*\|^2 - \text{tr}[(Z_P^\dagger Z_P) \beta_P^* \beta_P^{*T}] + \text{tr}[Z_{PC}^T (Z_P Z_P^T)^\dagger Z_{PC} \beta_{PC}^* \beta_{PC}^{*T}]$$

On a supposé que  $\mathbb{E}(\beta^* \beta^{*T}) = \sigma_{\beta^*}^2 I_E$ .

On a  $\sigma_\Phi^{-2} \mathcal{E}(\hat{y}_D) = \mathbb{E}(\|\beta^*\|^2) - \text{tr}[Z_P^\dagger Z_P \mathbb{E}(\beta_P^* \beta_P^{*T})] + \text{tr}[Z_{PC}^T Z_{PC}^T (Z_P Z_P^T)^\dagger \mathbb{E}(\beta_{PC}^* \beta_{PC}^{*T})]$

D'où  $(\sigma_\Phi \sigma_{\beta^*})^{-2} \mathcal{E}(\hat{y}_D) = E - \text{tr}(Z_P^\dagger Z_P) + \text{tr}[(Z_{PC}^T Z_{PC}^T) (Z_P Z_P^T)^\dagger]$

Or  $Z_E Z_E^T = Z_P Z_P^T + Z_{PC} Z_{PC}^T$ , d'où

$$\text{tr}[(Z_{PC}^T Z_{PC}^T) (Z_P Z_P^T)^\dagger] = \text{tr}[(Z_E Z_E^T) (Z_P Z_P^T)^\dagger] - \text{tr}[Z_P Z_P^T (Z_P Z_P^T)^\dagger] = \text{tr}[(Z_E Z_E^T) (Z_P Z_P^T)^\dagger] - rg(Z_P)$$

En effet la trace d'un projecteur est son rang, et  $rg(XX^T) = rg(X)$ . Donc de même  $\text{tr}(\Pi_{Z_P}) = rg(Z_P)$ .

Ainsi  $(\sigma_\Phi \sigma_{\beta^*})^{-2} \mathcal{E}(\hat{y}_D) = E - 2rg(Z_P) + \text{tr}[(Z_E Z_E^T) (Z_P Z_P^T)^\dagger]$

**Conjecture.** Expression de l'excès de risque asymptotique, cas du rang maximal (conjecture expérimental)

En supposant :  $\mathbb{E}(\beta^* \beta^{*T}) = \frac{\|\beta^*\|^2}{E} I_E$ ,  $rg(Z_P) = \min(P, N)$ , et  $(f_1, \dots, f_P)$  features orthonormées de moyenne nul.

$$\bar{\mathcal{E}}(\gamma, \delta) := \lim_{N \rightarrow +\infty, \frac{P}{N} \rightarrow \gamma, \frac{E}{N} \rightarrow \delta} \mathcal{E}(\hat{y}_D) = \sigma_\Phi^2 \left[ 1 - 2 \frac{\max(\gamma, 1)}{\delta} + \frac{\max(\gamma, 1)}{\delta} \left| \frac{1 - \delta}{1 - \gamma} \right| \right] \cdot \|\beta^*\|^2 \quad (9)$$

**Démonstration.** de cette conjecture

On se place dans le cadre  $\|\beta^*\|^2 = \text{tr}(\beta^* \beta^{*T}) = \text{constante}$  et  $rg(Z) = \min(N, P)$ . Ce choix vient du fait que l'erreur aléatoire sur les termes du développement en série diminue lorsque l'on prend plus de termes, et que l'ensemble des fonctions à approximer est tel que  $\mathbb{E}(\beta^*) = 0$ . On fixe  $\mathbb{V}(\beta_i^*) = \sigma_{\beta^*}^2$ . D'où  $\sigma_{\beta^*}^2 = \frac{\|\beta^*\|^2}{E}$  et ainsi on fait ensuite la limite  $E, P, N \rightarrow +\infty$ , avec  $\frac{E}{N} \rightarrow \delta$  et  $\frac{P}{N} \rightarrow \gamma$ .

Equation et conjecture: On suppose  $\frac{1}{\min(P, N)} \text{tr}[Z_E Z_E^T (Z_P Z_P^T)^\dagger] \sim \left| \frac{1 - \delta}{1 - \gamma} \right|$

- Régime sous-paramétré:

$$\sigma_{\Phi}^{-2} \mathcal{E}(\hat{y}_D) = \left[ 1 - \frac{2P}{E} + \frac{P}{E} \frac{1}{P} \text{tr}[Z_E Z_E^T (Z_P Z_P^T)^{\dagger}] \right] \cdot \|\beta^{\star}\|^2$$

$$\mathcal{E}(\hat{y}) \sim \sigma_{\Phi}^2 \left[ 1 - 2\frac{\gamma}{\delta} + \frac{\gamma}{\delta} \left| \frac{1-\delta}{1-\gamma} \right| \right] \cdot \|\beta^{\star}\|^2$$

- Régime sur-paramétré:

$$\sigma_{\Phi}^{-2} \mathcal{E}(\hat{y}_D) = \left[ 1 - \frac{2N}{E} + \frac{N}{E} \frac{1}{N} \text{tr}[Z_E Z_E^T (Z_P Z_P^T)^{\dagger}] \right] \cdot \|\beta^{\star}\|^2$$

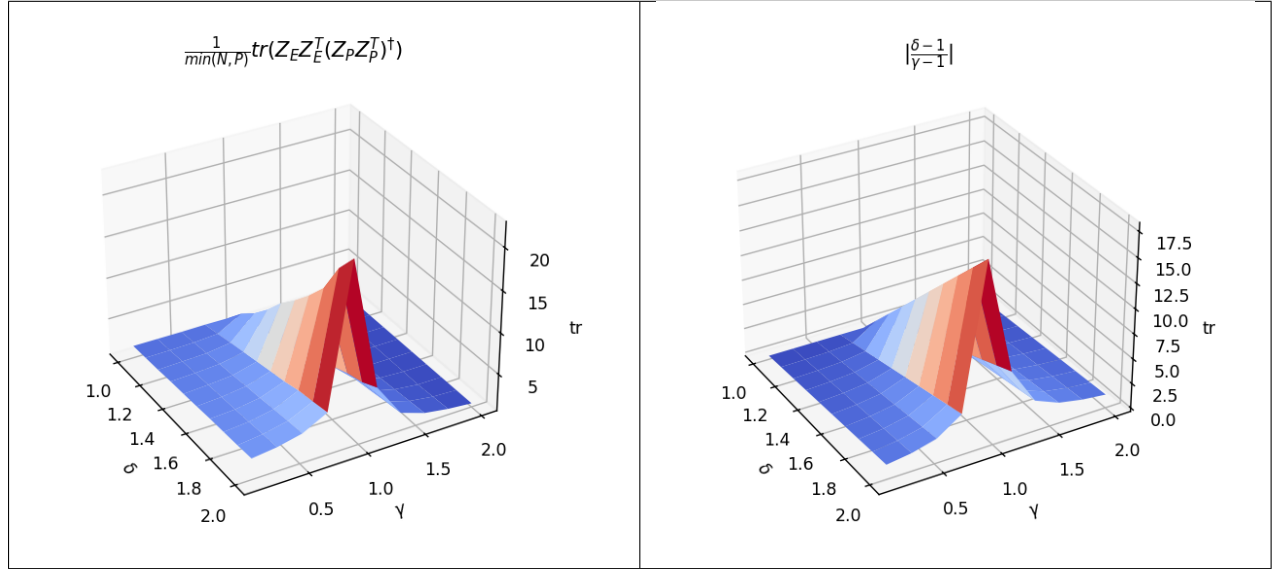
$$\mathcal{E}(\hat{y}) \sim \sigma_{\Phi}^2 \left[ 1 - 2\frac{1}{\delta} + \frac{1}{\delta} \left| \frac{1-\delta}{1-\gamma} \right| \right] \cdot \|\beta^{\star}\|^2$$

Pour tenter de prouver cela on pourrait tenter d'utiliser: Utiliser le lemme de Francis Bach page 8.

On a  $\text{tr}(A(\hat{\Sigma} + \lambda I)^{-1}) \sim \frac{\kappa(\lambda)}{\lambda} \text{tr}(A(\Sigma + \kappa(\lambda)I)^{-1})$  et faire  $\lambda \rightarrow 0$ . Pour  $\gamma < 1$  cela donne  $\text{tr}(A\hat{\Sigma}^{-1}) \sim \frac{1}{1-\gamma} \text{tr}(A\Sigma^{-1})$ . Pour  $\gamma > 1$  cela donne  $\text{tr}(A\hat{\Sigma}^{-1}) \sim \frac{1}{\gamma-1} \text{tr}(A(\Sigma + \sigma^2(\gamma-1)I)^{-1})$

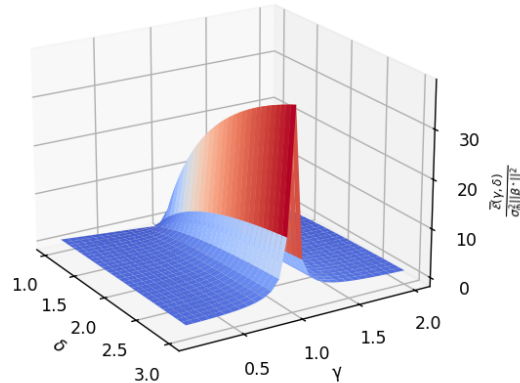
Et on peut aussi tenter d'utiliser la "distribution de Marchenko-Pastur" sur  $\frac{1}{N} Z_P Z_P^T$ .

Justification de la conjoncture: (avec loi normale tel que  $\sigma = 1$ , pour  $N = 500$ )



Remarque: Pour  $E = P$  on trouve alors  $\bar{\mathcal{E}}(\gamma, \delta) = \max(0, (1 - \frac{1}{\gamma}))$  et on n'a pas de double descente. Mais pour  $\delta > 1$  on en a une.

$$\frac{\bar{\mathcal{E}}(\gamma, \delta)}{\sigma_{\Phi}^2 \|\beta^{\star}\|^2} = \left[ 1 - 2\frac{\max(\gamma, 1)}{\delta} + \frac{\max(\gamma, 1)}{\delta} \left| \frac{1-\delta}{1-\gamma} \right| \right]$$





## 6.2 Minimisation par descente de gradient

Remarque :

On a avec  $\hat{\beta}_t$  paramètre issu d'une descente de gradient à  $t$  étapes:

$$\sigma_{\Phi}^{-2} \mathcal{E}(\hat{y}_{\hat{\beta}_t}, \beta^*) = \|\hat{\beta}_t - \beta^*\|^2 = \|\hat{\beta}_t - \hat{\beta}\|^2 + 2 \langle \hat{\beta}_t - \hat{\beta}, \hat{\beta} - \beta^* \rangle + \|\hat{\beta} - \beta^*\|^2$$

On peut donc s'intéresser au terme d'erreur apporté par la descente de gradient, par rapport à la solution du pseudo-inverse.

### 6.2.1 Descente de gradient simple

La descente de gradient a pas constant est donnée par:

$$\hat{\beta}_{t+1} = \hat{\beta}_t - \alpha \nabla \hat{\mathcal{R}}_{\mathcal{D}}(\hat{y}_{\hat{\beta}_t})$$

On a  $\hat{\mathcal{R}}_{\mathcal{D}}(\hat{y}) = \frac{1}{N} \|Y - Z\hat{\beta}\|_2^2$  et  $\nabla \hat{\mathcal{R}}_{\mathcal{D}}(\hat{y}) = \frac{2}{N} Z^T [Z\hat{\beta} - Y]$  ainsi  $\hat{\beta}_{t+1} = [I_p - \frac{2\alpha}{N} Z^T Z] \hat{\beta}_t + \frac{2\alpha}{N} Z^T Y$

On sait qu'avec des suites du type  $x_{n+1} = ax_n + b$  on a  $x_n = [\sum_{k=0}^{n-1} a^k]b + a^n x_0$

D'où

$$\hat{\beta}_t = \left( \sum_{k=0}^{t-1} [I_p - \frac{2\alpha}{N} Z^T Z]^k \right) \frac{2\alpha}{N} Z^T Y + [I_p - \frac{2\alpha}{N} Z^T Z]^t \hat{\beta}_0$$

On note  $\alpha' = \frac{2\alpha}{N}$ .

**Théorème 7.** Approximation par descente de gradient simple Avec  $\hat{\beta}_t$  le résultat de  $t$  descente de gradient de pas  $\alpha' < \sigma_{max}^{-2}$  et  $\hat{\beta}_0 = 0$ . Avec  $\sigma_{min} > 0$  la plus petite valeur singulière non-nulle de  $Z$ , on a:

$$\sigma_{max}^{-1} (1 - \alpha' \sigma_{max}^2)^t \|Y\|_2 \leq \|\hat{\beta}_t - \hat{\beta}\|_2 \leq \sigma_{min}^{-1} (1 - \alpha' \sigma_{min}^2)^t \|Y\|_2 \quad (10)$$

**Démonstration.** On s'attend classiquement à une convergence géométrique de la descente de gradient.

Avec  $R = \text{rg}(Z)$  et  $Z = U\Sigma V^T$  décomposition SVD tel que  $\Sigma = \begin{bmatrix} \Sigma_R & 0 \\ 0 & 0 \end{bmatrix}$ . On a  $Z^T Z = V\Sigma^T \Sigma V^T = V \begin{bmatrix} \Sigma_R^2 & 0 \\ 0 & 0 \end{bmatrix} V^T$ .

On note  $\alpha' = \frac{2\alpha}{N}$  D'où  $\hat{\beta}_t = V \begin{bmatrix} (\alpha' \Sigma_R^2)^{-1} [I_R - (I_R - \alpha' \Sigma_R^2)^t] & 0 \\ 0 & t I_{P-R} \end{bmatrix} \alpha' \begin{bmatrix} \Sigma_R & 0 \\ 0 & 0 \end{bmatrix} U^T Y + V \begin{bmatrix} (I_R - \alpha' \Sigma_R^2)^t & 0 \\ 0 & I_{P-R} \end{bmatrix} V^T \hat{\beta}_0$

$\hat{\beta}_t = V \begin{bmatrix} \Sigma_R^{-1} - \Sigma_R^{-2} (I_R - \alpha' \Sigma_R^2)^t \Sigma_R & 0 \\ 0 & 0 \end{bmatrix} U^T Y + V \begin{bmatrix} (I_R - \alpha' \Sigma_R^2)^t & 0 \\ 0 & I_{P-R} \end{bmatrix} V^T \hat{\beta}_0$

Si  $\hat{\beta}_0 = 0$  on a:

$$\hat{\beta}_t - \hat{\beta} = -V \begin{bmatrix} \Sigma_R^{-1} (I_R - \alpha' \Sigma_R^2)^t & 0 \\ 0 & 0 \end{bmatrix} U^T Y$$

Dans le cas général on a  $\hat{\beta}_t - \hat{\beta} = V \begin{bmatrix} (I_R - \alpha' \Sigma_R^2)^t & 0 \\ 0 & I_{P-R} \end{bmatrix} V^T [\hat{\beta}_0 - \hat{\beta}]$

Or on rappelle que  $\hat{\beta} = Z^\dagger Y$ . Si de plus  $\alpha < \frac{N}{2\|Z\|_2^2}$  alors:

$$\|\hat{\beta}_t - \hat{\beta}\|_2 \leq \rho(\Sigma_R^{-1} (I_R - \alpha' \Sigma_R^2)^t) \|Y\|_2$$

Remarque: En notant  $V = [v_1, \dots, v_P]$  on a alors  $\|\hat{\beta}_t - \hat{\beta}\|_2^2 = \sum_{i=1}^R |\sigma_i^{-1} (1 - \alpha' \sigma_i^2)^t|^2 \langle v_i, Y \rangle^2$

On peut étudier la fonction  $f(\sigma) = \sigma^{-1} (1 - \alpha' \sigma^2)^t$  pour être plus précis, et on a alors deux annulations de la dérivée en  $\sigma = \pm \alpha'^{-\frac{1}{2}}$ . On se place donc dans la zone de décroissance de  $f$ . Donc, sous nos hypothèses, on a  $\rho(\Sigma_R^{-1} (I_R - \alpha' \Sigma_R^2)^t) = \sigma_{min}^{-1} (1 - \alpha' \sigma_{min}^2)^t$ .

On a finalement le résultat attendu.

Remarque: On remarque que l'on retrouve un résultat proche de celui de la descente de gradient classique, ici  $\hat{\mathcal{R}}$  étant  $\mu = \frac{2}{N}$  fortement convexe.

**Corollaire 2.** *Résultat de la descente de gradient simple, cas du rang maximale*  
On suppose que  $\hat{\beta}_0 = 0$  et que  $\alpha$  constant le pas de la descente.

- Si  $P < N$  et  $\text{rg}(Z) = P$ :

$$\hat{\beta} - \hat{\beta}_t = (Z^T Z)^{-1} (I_P - \alpha' Z^T Z)^t Z^T Y \quad (11)$$

- Si  $P > N$  et  $\text{rg}(Z) = N$ :

$$\hat{\beta} - \hat{\beta}_t = Z^T (I_N - \alpha' Z Z^T)^t (Z Z^T)^{-1} Y \quad (12)$$

**Démonstration.** On s'attend à avoir une forme simple dans le cas du rang maximal

- Dans le régime sous-paramétré: (cas du rang maximale)

On a  $\hat{\beta}_t = (\sum_{k=0}^{t-1} [I_P - \alpha' Z^T Z]^k) \alpha' Z^T Y = (Z^T Z)^{-1} [I_P - (I_P - \alpha' Z^T Z)^t] Z^T Y = Z^\dagger Y - (Z^T Z)^{-1} (I_P - \alpha' Z^T Z)^t Z^T Y$

- Dans le régime sur-paramétré: (cas du rang maximale)

On a ici  $Z Z^T = U \Sigma_N^2 U^T$  et  $Z^\dagger Y - \hat{\beta}_t = V \begin{bmatrix} \Sigma_N^{-1} (I_N - \alpha' \Sigma_N^2)^t \\ 0 \end{bmatrix} U^T Y = V \begin{bmatrix} \Sigma_N \\ 0 \end{bmatrix} (U^T U) (I_N - \alpha' \Sigma_N^2)^t (U^T U) \Sigma_N^{-2} U^T Y = (V \Sigma^T U^T) (I_N - \alpha' U \Sigma_N^2 U^T)^t (U \Sigma_N^2 U^T)^{-1} Y$

D'où:  $\hat{\beta}_t = Z^\dagger Y - Z^T (I_N - \alpha' Z Z^T)^t (Z Z^T)^{-1} Y$

### 6.2.2 Descente de gradient à pas variable

La descente de gradient a pas variable est donnée par:

$$\hat{\beta}_{t+1} = \hat{\beta}_t - \alpha_t \nabla \hat{\mathcal{R}}_{\mathcal{D}}(\hat{y}_{\hat{\beta}_t})$$

Et donc :  $\hat{\beta}_{t+1} = [I_P - \frac{2\alpha_t}{N} Z^T Z] \hat{\beta}_t + \frac{2\alpha_t}{N} Z^T Y$ , on est dans le cas  $u_{n+1} = a_n u_n + b_n$ . Dans ce cas on a  $u_n = [\prod_{k=0}^{n-1} a_k] u_0 + \sum_{k=0}^{n-1} [\prod_{i=k+1}^{n-1} a_i] b_k$ .

D'où:

$$\hat{\beta}_t = \left[ \prod_{k=0}^{t-1} [I_P - \frac{2\alpha_k}{N} Z^T Z] \right] \hat{\beta}_0 + \sum_{k=0}^{t-1} \left[ \prod_{i=k+1}^{t-1} [I_P - \frac{2\alpha_i}{N} Z^T Z] \right] \frac{2\alpha_k}{N} Z^T Y$$

On suppose  $\hat{\beta}_0 = 0$  et on pose  $Z = U \Sigma V^T$  où  $R = \text{rg}(Z)$ ,  $\Sigma = \begin{bmatrix} \Sigma_R & 0 \\ 0 & 0 \end{bmatrix}$

Dans ce contexte en notant  $\alpha'_i = \frac{2\alpha_i}{N}$ , avec la décomposition SVD de  $Z = U \Sigma V^T$  on a:

$$\hat{\beta}_t = \sum_{k=0}^{t-1} (\prod_{i=k+1}^{t-1} [I_P - \frac{2\alpha_i}{N} Z^T Z]) \frac{2\alpha_k}{N} Z^T Y = V [\sum_{k=0}^{t-1} (\prod_{i=k+1}^{t-1} [I_P - \alpha'_i \Sigma^T \Sigma]) \alpha'_k] \Sigma^T U^T Y$$

$$\hat{\beta}_t = V \begin{bmatrix} \sum_{k=0}^{t-1} (\prod_{i=k+1}^{t-1} I_R - \alpha'_i \Sigma_R^2) \alpha'_k & 0 \\ 0 & (\sum_{k=0}^{t-1} \alpha'_k) I_{P-R} \end{bmatrix}^T \begin{bmatrix} \Sigma_R & 0 \\ 0 & 0 \end{bmatrix} U^T Y = V \begin{bmatrix} \Sigma_R \sum_{k=0}^{t-1} (\prod_{i=k+1}^{t-1} I_R - \alpha'_i \Sigma_R^2) \alpha'_k & 0 \\ 0 & 0 \end{bmatrix} U^T Y$$

$$\hat{\beta}_t = \hat{\beta} + V \begin{bmatrix} \Sigma_R \sum_{k=0}^{t-1} (\prod_{i=k+1}^{t-1} I_R - \alpha'_i \Sigma_R^2) \alpha'_k - \Sigma_R^{-1} & 0 \\ 0 & 0 \end{bmatrix} U^T Y$$

On va tenter de simplifier le problème en considérant un pas  $\alpha_t$  constant par morceaux et cela permettra d'obtenir une forme explicite de l'erreur, commençons par un cas simple.

Supposons que  $\forall t \leq t_1 - 1, \alpha_t = \underline{\alpha}_1$  et que  $\forall t > t_1 - 1, \alpha_t = \underline{\alpha}_2$ .

Alors:  $\forall t > t_1$

$$\begin{aligned}\hat{\beta}_t - \hat{\beta} &= V \begin{bmatrix} \Sigma_R(I_R - \underline{\alpha}'_2 \Sigma_R^2)^{t-t_1} \sum_{k=0}^{t_1-1} (I_R - \underline{\alpha}'_1 \Sigma_R^2)^{t_1-k-1} \underline{\alpha}'_1 + \Sigma_R \sum_{k=t_1}^{t-1} (I_R - \underline{\alpha}'_2 \Sigma_R^2)^{t-k-1} \underline{\alpha}'_2 - \Sigma_R^{-1} & 0 \\ 0 & 0 \end{bmatrix} U^T Y \\ \hat{\beta}_t - \hat{\beta} &= -V \begin{bmatrix} \Sigma_R^{-1} (I_R - \underline{\alpha}'_2 \Sigma_R^2)^{t-t_1} (I_R - \underline{\alpha}'_1 \Sigma_R^2)^{t_1} & 0 \\ 0 & 0 \end{bmatrix} U^T Y\end{aligned}$$

On suppose maintenant  $t_1, \dots, t_r$  changements de pas de descente de gradient. On a alors avec  $t_0 = 0$  et  $t_{r+1} = t - 1$ :  $\forall j \leq r \forall t \in [t_j, t_{j+1}[$ ,  $\alpha'_t = \underline{\alpha}'_j$ . On s'appuie sur l'expression générale par récurrence sur le nombre de phase, avec  $\beta_0 = 0$

**Théorème 8.** *Descente de gradient à pas constant par morceaux tel que  $\hat{\beta}_0 = 0, \forall t > t_r$*

$$\hat{\beta} - \hat{\beta}_t = V \begin{bmatrix} \Sigma_R^{-1} (I_R - \underline{\alpha}'_r \Sigma_R^2)^{t-t_r} \prod_{j=0}^{r-1} (I_R - \underline{\alpha}'_j \Sigma_R^2)^{t_{j+1}-t_j} & 0 \\ 0 & 0 \end{bmatrix} U^T Y \quad (13)$$

On remarque qualitativement avec cette expression à l'aide des variations de la fonction  $f(\sigma) = \sigma^{-1} \prod_{j=0}^r (1 - \gamma_j \sigma^2)^{\Delta t_j} = \sigma^{-1} g(\sigma)$ , tel que  $f(\sigma) \underset{\sigma \rightarrow 0}{\sim} 1/\sigma$ , et tel que  $f'(\sigma) = \frac{-g(\sigma)}{\sigma^2} [2 \sum_{j=0}^r \frac{\Delta t_j \gamma_j \sigma^2}{1 - \gamma_j \sigma^2} + 1] \underset{\sigma \rightarrow 0}{\sim} -1/\sigma^2$  et dont les points critiques sont les solutions de  $\sum_{j=0}^r \gamma_j \sigma^2 \Delta t_j (1 - \gamma_j \sigma^2)^{-1} = -1/2$  et les points  $\sigma = \pm \gamma_j^{-1/2}$ . Sous la condition  $\max(\gamma_j) \leq \sigma_{\max}^{-2}$ , la première équation n'a pas de solution et l'on se place dans la zone de décroissance de  $f$ , on peut donc se ramener à la plus petite et la plus grande des valeurs singulière de  $Z$  en utilisant le théorème 11 et le lemme:  $\sigma(A)_{\min} \|X\|_2 \leq \|AX\|_2 \leq \sigma(A)_{\max} \|X\|_2$

**Corollaire 3.** *Approximation par descente de gradient à pas variable*

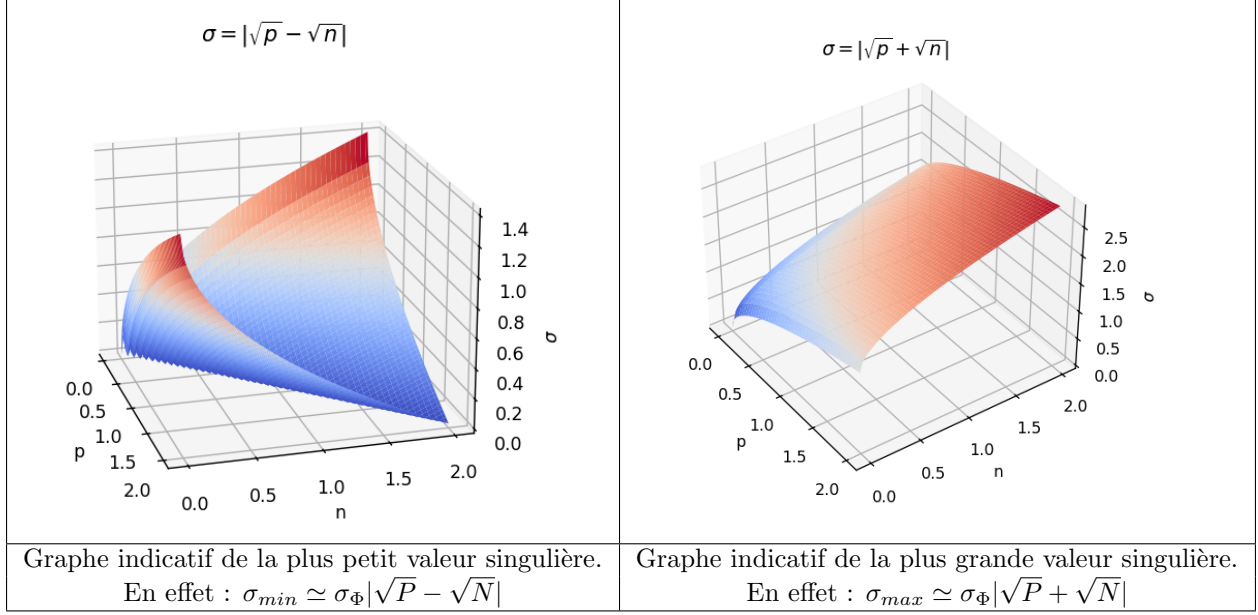
*Dans le cas où  $\hat{\beta}_0 = 0$  et  $\max(\underline{\alpha}'_j) < \sigma_{\max}^{-2}$ , en notant  $t_{r+1} := t > t_r$  on a:*

$$\sigma_{\max}^{-1} \prod_{j=0}^r (1 - \underline{\alpha}'_j \sigma_{\max}^2)^{t_{j+1}-t_j} \|Y\|_2 \leq \|\hat{\beta}_t - \hat{\beta}\|_2 \leq \sigma_{\min}^{-1} \prod_{j=0}^r (1 - \underline{\alpha}'_j \sigma_{\min}^2)^{t_{j+1}-t_j} \|Y\|_2$$

Ainsi, on a une erreur décroissante par rapport à  $t$  et  $\sigma$ . Donc  $\sigma_{\min}$  ayant une courbe en U (voir section suivante) on a bien une courbe en U inversée pour l'erreur en fonction du quotient  $P \backslash N$ . Ainsi l'erreur apporté par la descente de gradient présente bien un apport au phénomène de double descente. De plus, on remarquera qu'ici l'inégalité obtenue est optimal au sens que la seule inégalité utilisée dans la preuve est  $\sigma(A)_{\min} \|X\|_2 \leq \|AX\|_2 \leq \sigma(A)_{\max} \|X\|_2$ .

Remarque: Ici, la condition sur le pas est compatible avec un cadre appliqué mais ne permet pas de réaliser une descente de gradient à pas optimal.

### 6.2.3 Analyse de la plus petite et plus grande valeur singulière



”On the limit of the largest eigenvalue” [10] et ”Marchenko–Pastur distribution” pour  $N \rightarrow +\infty$  et  $P \setminus N \rightarrow \gamma$ . Dans le cadre de notre base de features orthonormés, en notant  $\mathbb{V}(\Phi_P(x)) = \sigma_\Phi^2$ , on a alors  $\sigma_{min} \simeq \sigma_\Phi |\sqrt{P} - \sqrt{N}|$  qui présente bien une courbe en U, et  $\sigma_{max} \simeq \sigma_\Phi |\sqrt{P} + \sqrt{N}|$ .

### 6.2.4 Descente de gradient stochastique

La descente de gradient stochastique sur les coordonnées du gradient est donnée par:

$$\hat{\beta}_{t+1} = \hat{\beta}_t - \alpha_t D_t \nabla \hat{\mathcal{R}}_{\mathcal{D}_t}(\hat{y}_{\hat{\beta}_t})$$

où  $J_t = \text{Diag}(\delta_{1,t} \cdots \delta_{P,t})$  et  $\mathcal{D}_t \subset \mathcal{D}$ . Ce sont des variables aléatoires.  $J_t$  permet de sélectionner des composantes du gradient à calculer et  $\mathcal{D}_t$  permet de calculer le gradient sur une partie de l'échantillon (batch). Or  $\mathcal{D}_t \subset \mathcal{D}$ ,  $\mathcal{D}_t = \{(x_{n_t(b)}, y_{n_t(b)}) \in \mathcal{X} \times \mathcal{Y}, b \in [1, B]\}$ . Et  $n_t : [1, B] \rightarrow [1, N]$  une injection. On peut alors poser  $D_t := [\delta_{n_t(i), j}] \in \mathcal{M}_{B, N}(\mathbb{R})$ .

On trouve alors  $\hat{\beta}_{t+1} = [I_P - \alpha'_t J_t Z^T D_t^T D_t Z] \hat{\beta}_t + \alpha'_t J_t Z^T D_t^T D_t Y$ .

On pose:  $P_t := D_t^T D_t = \text{Diag}(\delta_{i \in \text{Im}(n_t)})$  D'où:

$$\hat{\beta}_t = \left[ \prod_{k=0}^{t-1} (I_P - \alpha'_k J_k Z^T P_k Z) \right] \hat{\beta}_0 + \sum_{k=0}^{t-1} \left[ \prod_{i=k+1}^{t-1} (I_P - \alpha'_i J_i Z^T P_i Z) \right] \alpha'_k J_k Z^T P_k Y$$

Ainsi en prenant  $J \hookrightarrow \text{Diag}(\mathcal{U}([1, P]))$  et  $\mathcal{D}_t \hookrightarrow \mathcal{U}(\mathcal{D}^B)$ , indépendantes deux à deux on a alors:

On a donc les mêmes résultats que précédemment, mais avec un facteur  $\frac{1}{P}$  et  $\frac{B}{N}$  et cette fois sur  $\|\mathbb{E}_J(\hat{\beta}_t) - \hat{\beta}\|_2$ .

$$\mathbb{E}_{\mathcal{D}, J}(\hat{\beta}_t) = \left[ \prod_{k=0}^{t-1} (I_P - \alpha'_k \frac{B}{NP} Z^T Z) \right] \hat{\beta}_0 + \sum_{k=0}^{t-1} \left[ \prod_{i=k+1}^{t-1} (I_P - \alpha'_i \frac{B}{NP} Z^T Z) \right] \alpha'_k \frac{B}{NP} Z^T Y$$

## 6.3 Expressions théorique sur les MLP

$\hat{y}_\beta(x) = W_L \circ \sigma \circ \cdots \circ \sigma \circ W_1(x)$  On a

$W_l(x_l) = A_l x_l + b_l$  Et  $\sigma(x) = \max(0, x)$  fonction ReLU tq  $\sigma'(x) = \mathbf{1}_{x \geq 0}$

Cas particulier: Dans le cas suivant,  $\hat{y}_\beta(x) = W_2 \circ \sigma \circ W_1(x) = A_2 \sigma(A_1 x + b_1) + b_2$ ,  $\hat{y}_\beta : \mathbb{R}^D \rightarrow \mathbb{R}$ . On a ici :  $A_1 \in \mathcal{M}_{P, D}(\mathbb{R})$  et  $A_2 \in \mathcal{M}_{1, P}(\mathbb{R})$  D'où avec :  $\hat{\mathcal{R}}_{\mathcal{D}}(\hat{y}_{\hat{\beta}}) = \frac{1}{N} \sum_{i=1}^N (\hat{y}_{\hat{\beta}}(x_i) - y_i)^2$  on a :

$$\text{Avec } \beta = \begin{bmatrix} A_1 \\ b_1 \\ A_2 \\ b_2 \end{bmatrix} \text{ on a } \nabla_{\beta} \hat{\mathcal{R}}(\hat{y}_{\hat{\beta}}) = \begin{bmatrix} \nabla_{A_1} \hat{\mathcal{R}}(\hat{y}_{\hat{\beta}}) \\ \nabla_{b_1} \hat{\mathcal{R}}(\hat{y}_{\hat{\beta}}) \\ \nabla_{A_2} \hat{\mathcal{R}}(\hat{y}_{\hat{\beta}}) \\ \nabla_{b_2} \hat{\mathcal{R}}(\hat{y}_{\hat{\beta}}) \end{bmatrix} = \frac{2}{N} \begin{bmatrix} \sum_{i=1}^N x_i (\hat{A}_2 \odot \sigma'(\hat{A}_1 x_i + \hat{b}_1)^T) (\hat{y}_{\hat{\beta}}(x_i) - y_i) \\ \sum_{i=1}^N (\hat{A}_2 \odot \sigma'(\hat{A}_1 x_i + \hat{b}_1)^T) (\hat{y}_{\hat{\beta}}(x_i) - y_i) \\ \sum_{i=1}^N \sigma(\hat{A}_1 x_i + \hat{b}_1) (\hat{y}_{\hat{\beta}}(x_i) - y_i) \\ \sum_{i=1}^N (\hat{y}_{\hat{\beta}}(x_i) - y_i) \end{bmatrix}$$

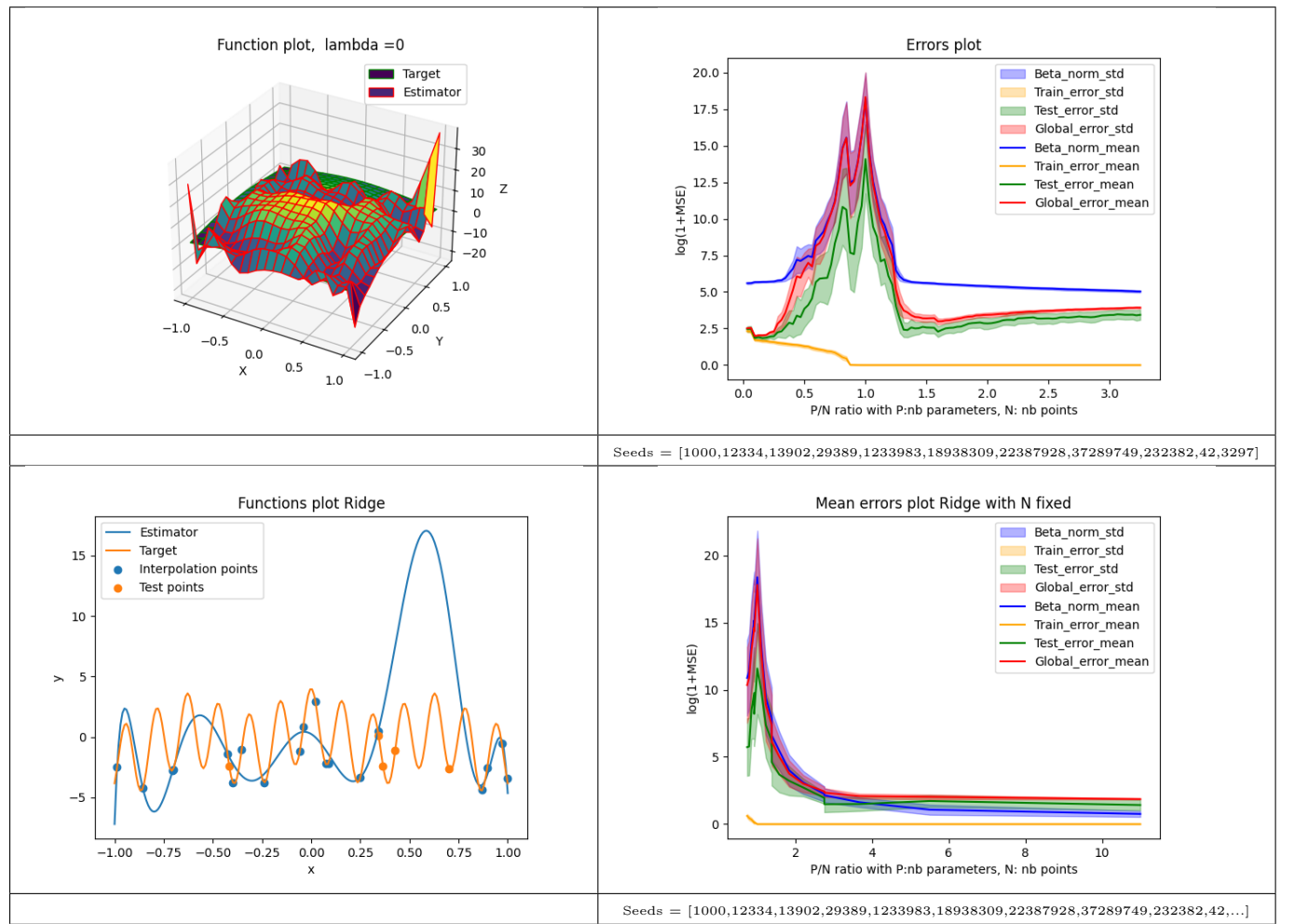
Avec  $\odot$  le produit d'Hadamard.

Et on a :  $\hat{\beta}_{t+1} = \hat{\beta}_t - \alpha \nabla_{\beta} \hat{\mathcal{R}}(\hat{y}_{\hat{\beta}_t})$ . On comprend donc qu'il n'est pas aisé d'obtenir une expression manipulable théoriquement.

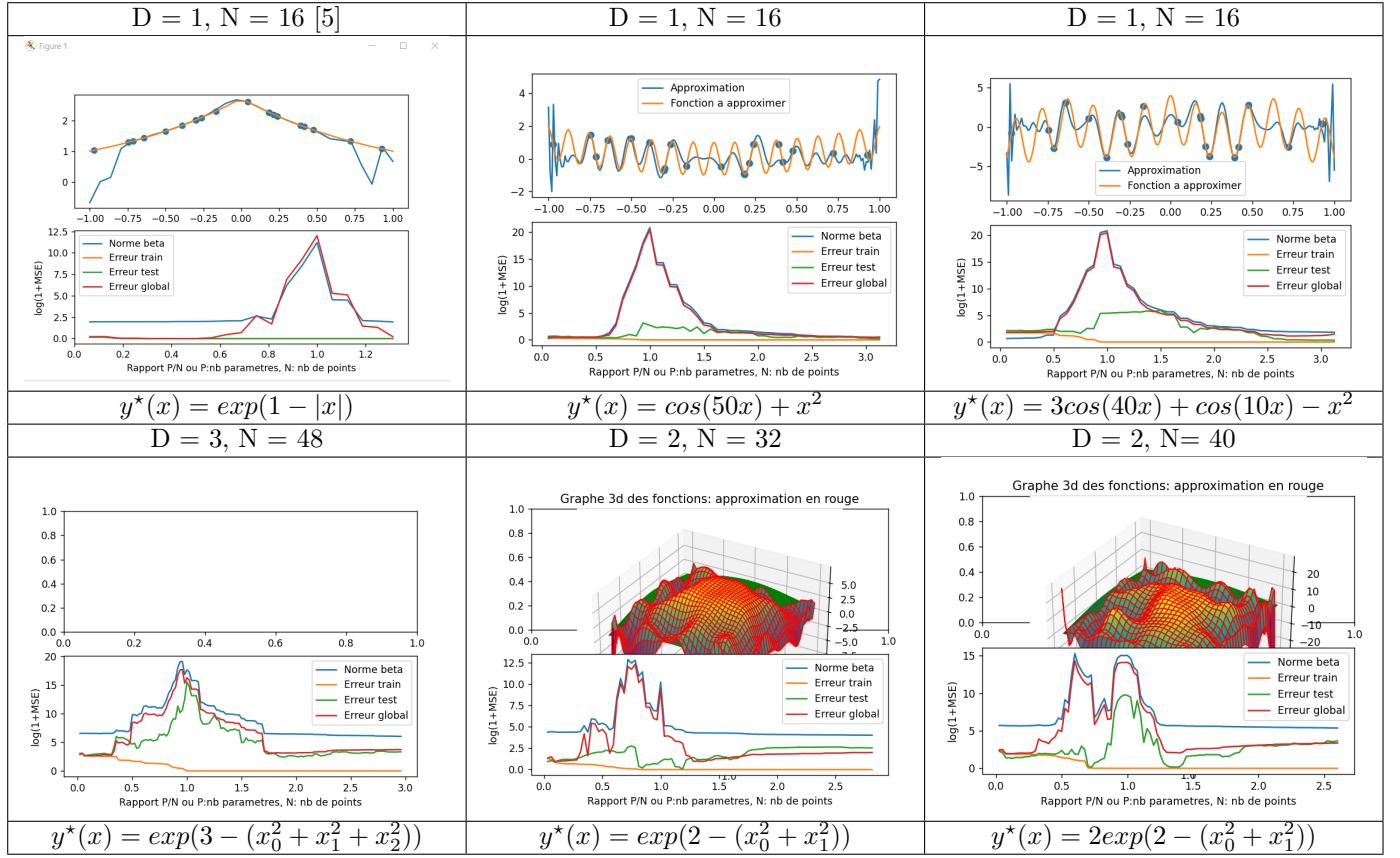
## 7 Résultats expérimentaux

### 7.1 Régression polynomiale

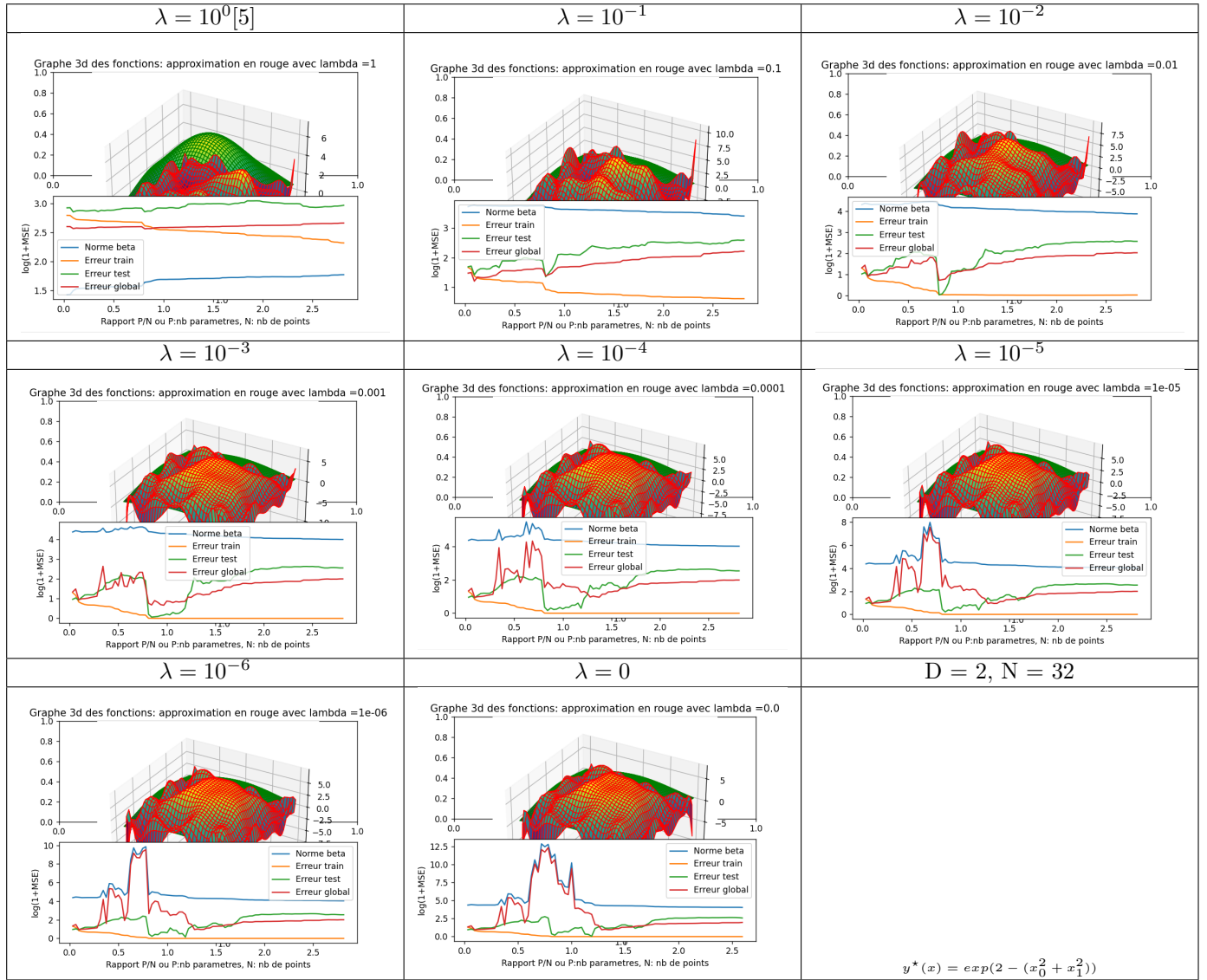
Demo en ligne [4] et github associé [5].



a, nexample = 1, 1, typepolynome = 2, D, Deg= 2,13, n = 20,M = 40 ,r = 0.2, Lambda = 0



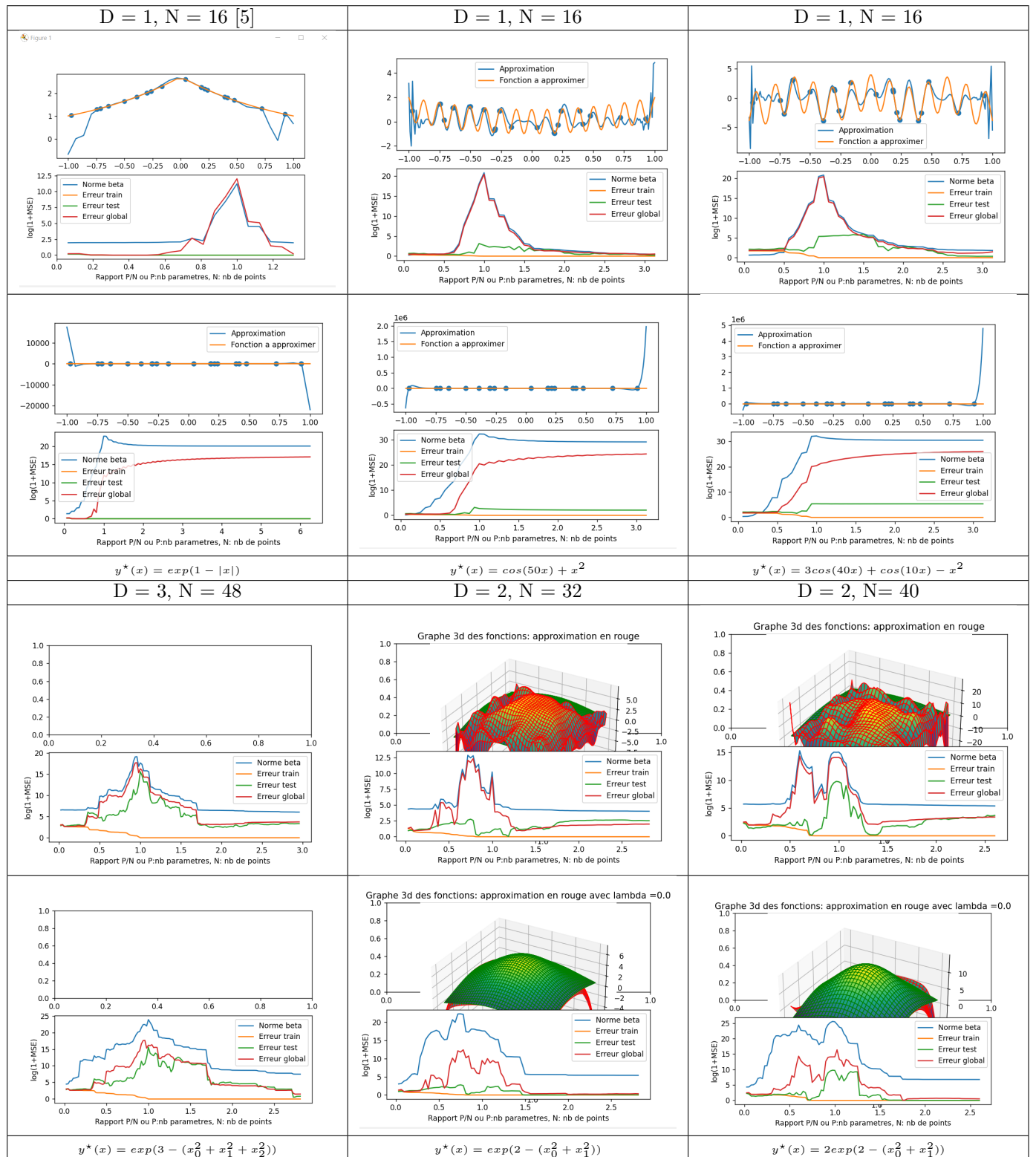
Résultats exhibant un phénomène de double descente en dimension 1, 2 et 3 pour des features polynomiales.



Résultats exhibant l'influence d'un facteur de régulation sur l'apparition du phénomène de double descente.

On remarque que la régulation permet d'empêcher totalement la venue d'un phénomène de double descente par rapport au nombre de paramètre. En effet, on peut considérer que la réelle complexité réside dans la norme du paramètre  $\beta$ . Or, on a une descente simple par rapport à ce paramètre. D'où le résultat.



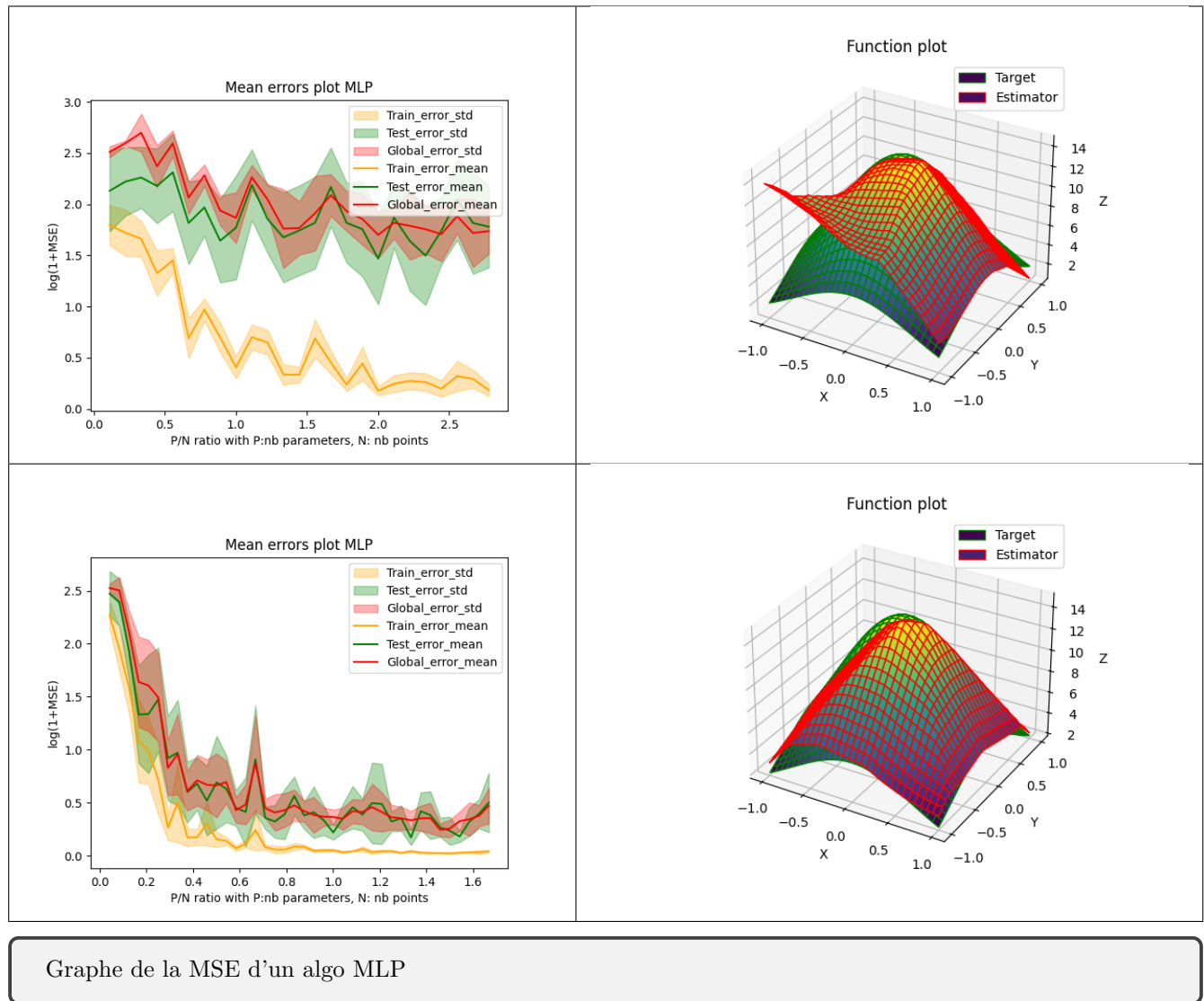


Résultats exhibant l'influence d'une base orthonormalisée sur la double descente. Première ligne: Base orthonormalisée sur l'espace, Deuxième ligne: Base canonique

On remarque qu'orthonormaliser l'espace permet s'accélérer l'avenue du phénomène de double descente en faible dimension.



## 7.2 Multilayer Perceptron (MLP)



## 8 Détails numériques

### References

- [1] Francis Bach. “High-dimensional analysis of double descent for linear regression with random projections”. In: *SIAM Journal on Mathematics of Data Science* 6.1 (2024), pp. 26–50.
- [2] Mikhail Belkin, Daniel Hsu, and Ji Xu. “Two models of double descent for weak features”. In: *SIAM Journal on Mathematics of Data Science* 2.4 (2020), pp. 1167–1180.
- [3] Mikhail Belkin et al. “Reconciling modern machine-learning practice and the classical bias–variance trade-off”. In: *Proceedings of the National Academy of Sciences* 116.32 (2019), pp. 15849–15854.
- [4] Emmett Haddad. *Demo Double Descente*. <https://ipolcore.ipol.im/demo/clientApp/demo.html?id=77777000515>. [Online]. 2024.
- [5] Emmett Haddad. *Github Double Descente*. <https://github.com/EmettGabrielH/Double-descente---Emett-Haddad>. [Online]. 2024.
- [6] Ilja Kuzborskij et al. “On the role of optimization in double descent: A least squares study”. In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 29567–29577.

- [7] Zhenyu Liao, Romain Couillet, and Michael W Mahoney. “A random matrix analysis of random fourier features: beyond the gaussian kernel, a precise phase transition, and the corresponding double descent”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 13939–13950.
- [8] Rylan Schaeffer et al. “Double descent demystified: Identifying, interpreting & ablating the sources of a deep learning puzzle”. In: *arXiv preprint arXiv:2303.14151* (2023).
- [9] MM Wolf. “Mathematical foundations of supervised learning (growing lecture notes)”. In: (2018).
- [10] Yong-Qua Yin, Zhi-Dong Bai, and Pathak R Krishnaiah. “On the limit of the largest eigenvalue of the large dimensional sample covariance matrix”. In: *Probability theory and related fields* 78 (1988), pp. 509–521.

## 9 Appendices

## 9.1 Pseudo-code

### 9.1.1 Modèle linéaire et pénalisé

---

**Algorithm 1:** Generation of the orthonormal polynomial basis (Gram- Schmidt)

---

```

1 function generate_orthonormal_basis(D, D', C)
  Input D, D', C:  $D \in \mathbb{N}^*$ ,  $D' \in \mathbb{N}^*$ ,  $C \subset \mathbb{R}^D$ 
2  BasisD' = [ $\prod_{d=1}^D X_d^{\alpha_d}$ ,  $\sum \alpha_d \leq D'$ ]
3  P = LENGHT(BasisD')
4  for 1 ≤ p ≤ P do
5     $f'_p = \text{Basis}_{D'}[p] - \sum_{i=1}^{p-1} [\int_C \text{Basis\_ortho}_{D'}[i] \cdot \text{Basis}_{D'}[p]] \times \text{Basis\_ortho}_{D'}[i]$ 
6     $\text{Basis\_ortho}_{D'}[p] = \frac{f'_p}{\|f'_p\|_2}$ 
7  return Basis_orthoD'

```

---



---

**Algorithm 2:** Dataset initialisation

---

```

1 function dataset_initialisation(f, C, M, ratio_data)
  Input y, C, M, ratio_data:  $y : \mathbb{R}^D \rightarrow \mathbb{R}$ ,  $C = [[a_d, b_d], 1 \leq d \leq D]$ , ratio_data ∈ [0, 1]
2   $U = \mathcal{U}([0, 1]^{(M,D)})$ 
3   $X = \text{Diag}(a_d) + U \text{Diag}(b_d - a_d)$ 
4   $Y = y(X)$ 
5   $X_{train}, X_{test}, Y_{train}, Y_{test} = \text{test\_split}(X, Y, \text{ratio\_data})$ 
6  return Xtrain, Xtest, Ytrain, Ytest

```

---



---

**Algorithm 3:** Ridge Regression

---

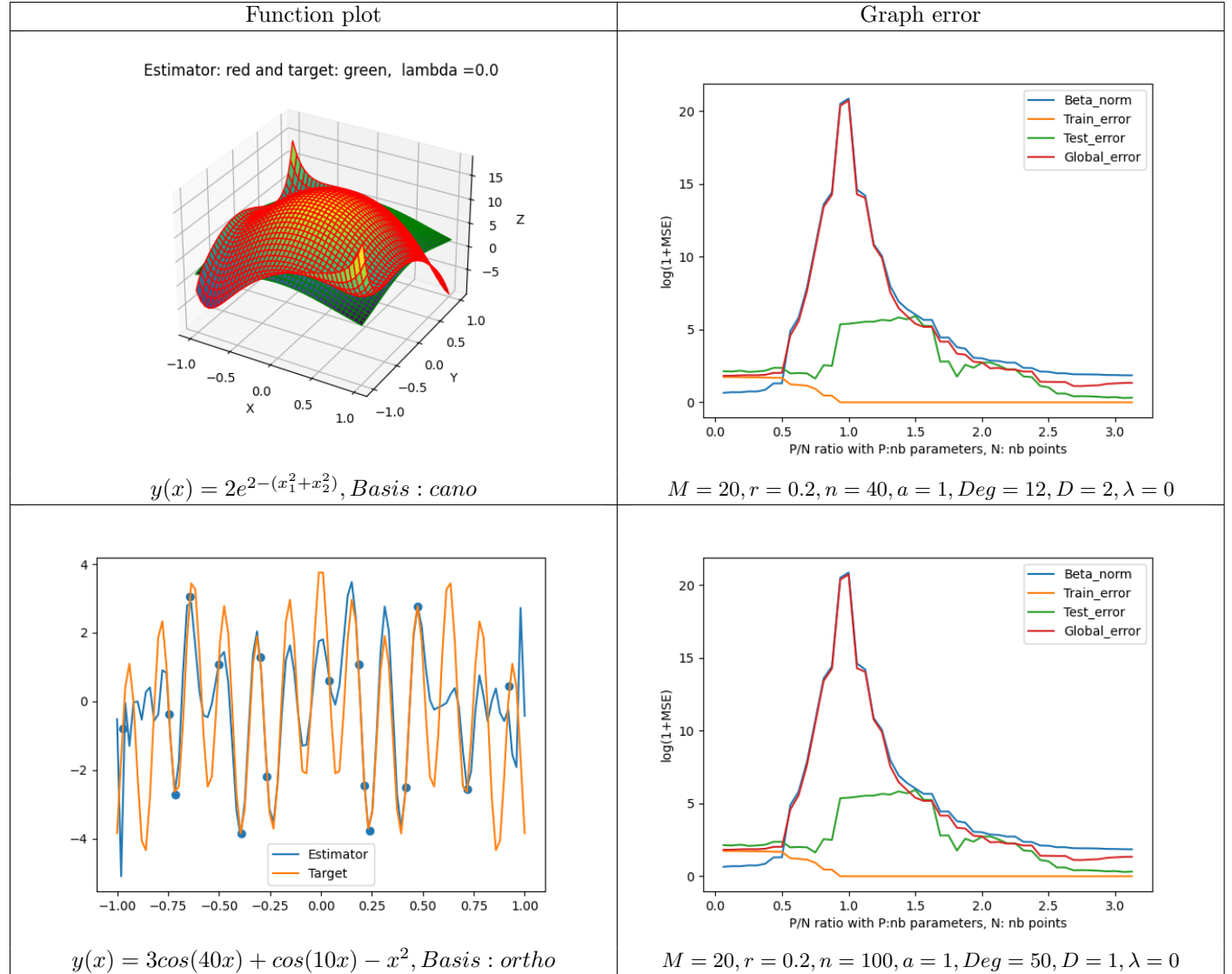
```

1 function ridge_regression(Pmin, Pmax, Features, λ, Data, Data_global)
2  Xtrain, Xtest, Ytrain, Ytest = Data
3  Xglobal, Yglobal = Data_global
4  for Pmin ≤ p ≤ Pmax do
5     $\Phi_p = [f_p, 1 \leq p \leq P]^T$ 
6     $Z_p = \Phi_p^T(X_{train})$ 
7     $\hat{\beta}_p = \begin{bmatrix} Z_p \\ \sqrt{N\lambda}I_p \end{bmatrix}^\dagger \begin{bmatrix} Y_{train} \\ O_p \end{bmatrix}$ 
8    Train_error[p] = log(1 + MSE(Ytrain, Zpβp))
9    Test_error[p] = log(1 + MSE(Ytest, ΦpT(Xtest)βp))
10   Global_error[p] = log(1 + MSE(Yglobal, ΦpT(Xglobal)βp))
11   Beta_norm[p] = log(1 + ||βp||22)
12    $\hat{y} = \Phi_p \cdot \hat{\beta}_{P_{max}}$ 
13  return Train_error, Test_error, Beta_norm, Global_error,  $\hat{y}$ 

```

---

Examples: (random seed : 23334)



### 9.1.2 Modèle MLP

---

#### Algorithm 4: MLP Gradient Descent

---

```

1 function MLP_Gradient_Descent( $P_{\min}, P_{\max}, \text{Epochs}, \alpha, \text{Data}, \text{Data\_global}$ )
2    $X_{\text{train}}, X_{\text{test}}, Y_{\text{train}}, Y_{\text{test}} = \text{Data}$ 
3    $X_{\text{global}}, Y_{\text{global}} = \text{Data\_global}$ 
4   for  $P_{\min} \leq p \leq P_{\max}$  do
5      $MLP = \text{Affine}(1, P) \circ \sigma \circ \text{Affine}(P, D)$ 
6      $MLP.\text{fit}(X_{\text{train}}, Y_{\text{train}}, \text{Epochs}, \alpha)$ 
7      $\text{Train\_error}[p] = \log(1 + \text{MSE}(Y_{\text{train}}, MLP(X_{\text{train}})))$ 
8      $\text{Test\_error}[p] = \log(1 + \text{MSE}(Y_{\text{test}}, MLP(X_{\text{test}})))$ 
9      $\text{Global\_error}[p] = \log(1 + \text{MSE}(Y_{\text{test}}, MLP(X_{\text{global}})))$ 
10   $\hat{y} = MLP(P_{\max})$ 
11  return Train_error, Test_error, Global_error,  $\hat{y}$ 

```

---

---

**Algorithm 5:** MLP Gradient Descent FIT
 

---

```

1 function MLP.fit( $X_{\text{train}}, Y_{\text{train}}, \text{Epochs}, \alpha$ )
2   for  $1 \leq \text{epoch} \leq \text{Epochs}$  do
3     for  $(x_{\text{train}}, y_{\text{train}}) \in \mathcal{D}$  do
4        $\text{score\_gradient} = \text{MLP}(x_{\text{train}}) - y_{\text{train}}$ 
5        $\text{backpropagation}(\text{score\_gradient})$ 
6      $\text{update}(\alpha)$ 

```

---

