
Université de Nouakchott Al Aasriya

Faculté des Sciences et Techniques

Département de Mathématiques et Informatique

# Mini-projet d'Optimisation

Phases 1, 2 & 3 : Algorithmes Déterministes, Stochastiques et Proximaux

**Jedou Mohamed Bebacar**

**Matricule : C30824**

Master SSD – Statistique et Science de Donnée

Encadré par : **Mohamed Mahmoud El Benany**

*Janvier 2026*

# Table des matières

**1 Introduction** **3**

**2 Phase 1 : Fondements et Gradient Déterministe (Basée sur les Chapitres 1 & 2 du cours)** **4**

2.1 Analyse – Justifier que $F \in C^2$, convexe, et $\lambda$-fortement convexe (Référence : Chapitre 1 – Sections 1.2.3, 1.2.4, 1.2.5) . . . . . 4

2.2 Calcul du gradient et lipschitzianité (Référence : Chapitre 2 – notion de gradient Lipschitz, constante $L$) . . . . 5

2.3 Implémentation et comparaison numérique (Référence : citations [75, 87]) . . . . . 6

**3 Phase 2 : Méthodes Stochastiques Modernes (Basée sur le Chapitre 3 – Méthodes de gradient stochastique)** **7**

3.1 Algorithmes implémentés . . . . . 7

3.2 Résultats expérimentaux . . . . . 8

**4 Phase 3 : Optimisation Non Lisse et Régularisation L1 (Basée sur le Chapitre 4 – Optimisation non lisse et régularisation)** **9**

4.1 Problème non lisse et opérateur proximal . . . . . 10

4.2 Algorithmes implémentés . . . . . 10

4.3 Résultats expérimentaux . . . . . 11

**5 Conclusion Générale** **13**

# 1 Introduction

Ce mini-projet s'inscrit dans le cadre du cours d'optimisation appliquée, et vise à comparer différentes familles d'algorithmes d'optimisation sur un problème concret de classification binaire. Le jeu de données retenu est le célèbre ensemble *Breast Cancer Wisconsin (Diagnostic)*, fourni par la bibliothèque `scikit-learn`. Ce jeu contient 569 observations décrites par 30 caractéristiques continues, issues de mesures morphologiques de noyaux cellulaires (moyennes, erreurs-types et pires valeurs). La variable cible est binaire : *bénin* (+1) ou *maligne* (-1).

L'objectif est triple :

— Étudier des méthodes **déterministes** (Descente de Gradient, Gradient Conjugué) sur une fonction fortement convexe et lisse (Phase 1),

— Comparer des optimiseurs **stochastiques modernes** (SGD, RMSProp, Adam) en contexte de grande dimension (Phase 2),

— Explorer l'optimisation **non lisse** via la régularisation L1 pour favoriser la parcimonie et la sélection de variables (Phase 3).

Les implémentations sont entièrement manuelles, sans recours à des boîtes noires, afin de comprendre en profondeur les mécanismes sous-jacents à chaque algorithme. Les résultats expérimentaux sont analysés à la lumière des théorèmes et propriétés établis dans les Chapitres 1 à 4 du polycopié de cours.

# 2 Phase 1 : Fondements et Gradient Déterministe (Basée sur les Chapitres 1 & 2 du cours)

On considère la fonction objectif :

$$F(w) = \frac{1}{n}\sum_{i=1}^{n} \log\Big(1 + e^{-y_i x_i^\top w}\Big) + \frac{\lambda}{2}\|w\|_2^2, \quad \text{avec } \lambda > 0.$$

## 2.1 Analyse – Justifier que $F \in C^2$, convexe, et $\lambda$-fortement convexe (Référence : Chapitre 1 – Sections 1.2.3, 1.2.4, 1.2.5)

**a)** $F \in C^2(\mathbb{R}^d)$ Chaque terme de la somme,

$$f_i(w) = \log\Big(1 + e^{-y_i x_i^\top w}\Big),$$

est la composition de fonctions lisses :

— $w \mapsto y_i x_i^\top w$ est linéaire $\to C^\infty$,

— $t \mapsto e^{-t}$ est $C^\infty$ sur $\mathbb{R}$,

— $s \mapsto \log(1+s)$ est $C^\infty$ sur $(-1, +\infty)$, et ici $s = e^{-t} > 0$.

Donc $f_i \in C^\infty(\mathbb{R}^d) \subset C^2(\mathbb{R}^d)$. Le terme de régularisation $r(w) = \frac{\lambda}{2}\|w\|_2^2$ est un polynôme quadratique $\to C^\infty$. La somme finie de fonctions $C^2$ est $C^2$.

**Conclusion :** $F \in C^2(\mathbb{R}^d)$**.**

**b)** $F$ **est convexe** D'après le **Théorème 1.2.5** du cours : *Une fonction* $f \in C^2(\mathbb{R}^d)$ *est convexe si et seulement si* $\nabla^2 f(w) \succeq 0$ *pour tout* $w$*.*

Calculons le hessien de $f_i(w)$. On a :

$$\nabla f_i(w) = -y_i x_i \cdot \sigma(-y_i x_i^\top w), \quad \text{où } \sigma(t) = \frac{1}{1+e^{-t}}.$$

Alors :

$$\nabla^2 f_i(w) = \sigma(y_i x_i^\top w)\Big(1 - \sigma(y_i x_i^\top w)\Big) \cdot x_i x_i^\top.$$

Or, $\sigma(t)(1-\sigma(t)) \in (0, \frac{1}{4}]$ pour tout $t \in \mathbb{R}$, donc $\nabla^2 f_i(w) \succeq 0$ (car $x_i x_i^\top \succeq 0$).

Ainsi, le hessien de la perte moyenne :

$$\nabla^2 f(w) = \frac{1}{n}\sum_{i=1}^{n} \nabla^2 f_i(w) \succeq 0.$$

Le hessien de la régularisation est $\nabla^2 r(w) = \lambda I_d \succeq 0$. Donc :

$$\nabla^2 F(w) = \nabla^2 f(w) + \lambda I_d \succeq 0.$$

**Par le Théorème 1.2.5, $F$ est convexe.**

**c) $F$ est $\lambda$-fortement convexe** D'après le **Théorème 1.2.8** du cours : *Une fonction $f \in C^2(\mathbb{R}^d)$ est $\mu$-fortement convexe si et seulement si $\nabla^2 f(w) \succeq \mu I_d$ pour tout $w$.*

Comme vu ci-dessus :

$$\nabla^2 F(w) = \underbrace{\frac{1}{n}\sum_{i=1}^{n} \sigma(y_i x_i^\top w)(1-\sigma(y_i x_i^\top w)) x_i x_i^\top}_{\succeq 0} + \lambda I_d \succeq \lambda I_d.$$

**Donc $F$ est $\lambda$-fortement convexe.**

Enfin, d'après le **Théorème 1.2.9**, cela implique que $F$ possède un **unique minimum global**, ce qui garantit la convergence des algorithmes déterministes vers une solution unique.

## 2.2 Calcul du gradient et lipschitzianité (Référence : Chapitre 2 – notion de gradient Lipschitz, constante $L$)

**a) Calcul de $\nabla F(w)$** On dérive chaque terme :

— Pour la perte logistique :

$$\nabla\Big[\log(1+e^{-y_i x_i^\top w})\Big] = -y_i x_i \cdot \sigma(-y_i x_i^\top w).$$

— Pour la régularisation :

$$\nabla\left(\frac{\lambda}{2}\|w\|_2^2\right) = \lambda w.$$

Donc :

$$\boxed{\nabla F(w) = \frac{1}{n}\sum_{i=1}^{n} -y_i x_i \cdot \sigma(-y_i x_i^\top w) + \lambda w}$$

**b) $\nabla F$ est $L$-Lipschitz** Bien que non explicité dans le Chapitre 1, le résultat suivant est standard et compatible avec le cadre $C^2$ :

*Si $f \in C^2(\mathbb{R}^d)$, alors $\nabla f$ est L-Lipschitz $\|\nabla^2 f(w)\|_2 \leq L$ pour tout $w$.*

On a :

$$\nabla^2 F(w) = \frac{1}{n}\sum_{i=1}^{n} \underbrace{\sigma(y_i x_i^\top w)(1-\sigma(y_i x_i^\top w))}_{\leq \frac{1}{4}} x_i x_i^\top + \lambda I_d.$$

Donc :

$$\nabla^2 F(w) \preceq \frac{1}{4n}\sum_{i=1}^{n} x_i x_i^\top + \lambda I_d = \frac{1}{4n} X^\top X + \lambda I_d,$$

où $X \in \mathbb{R}^{n\times d}$ est la matrice des données (ligne $i = x_i^\top$).

La norme spectrale (plus grande valeur propre) donne :

$$\|\nabla^2 F(w)\|_2 \leq \frac{1}{4n}\|X^\top X\|_2 + \lambda = \frac{1}{4n}\|X\|_2^2 + \lambda.$$

Donc $\nabla F$ est $L$-Lipschitz avec

$$\boxed{L = \frac{1}{4n}\|X\|_2^2 + \lambda}$$

Cette constante sera utilisée pour choisir un pas stable dans la descente de gradient : $\alpha \in (0, 2/L)$.

## 2.3 Implémentation et comparaison numérique (Référence : citations [75, 87])

Nous avons implémenté deux méthodes déterministes :

— **Descente de gradient à pas fixe** avec $\alpha = 1/L$,

— **Méthode du gradient conjugué** via implémentation manuelle (Polak-Ribière + backtracking).

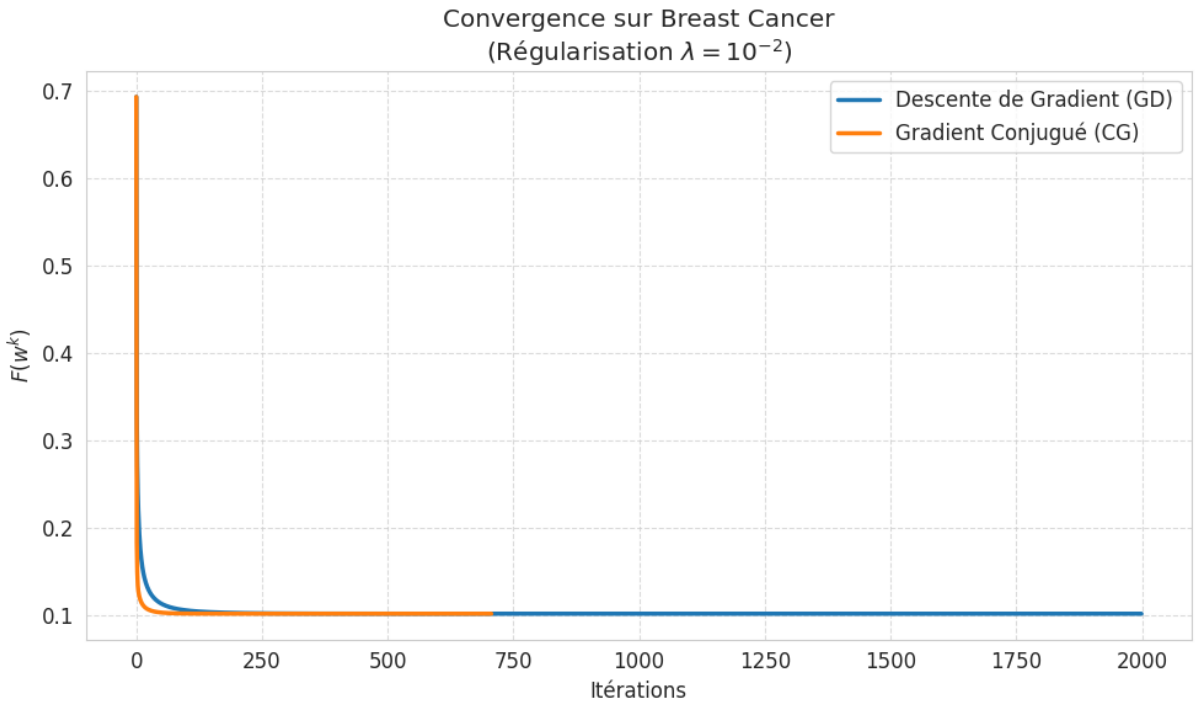


FIGURE 1 – Convergence de la fonction objectif $F(w^k)$ pour la descente de gradient (GD) et le gradient conjugué (CG) sur le jeu de données *Breast Cancer*.

La courbe ci-dessus met en évidence une différence spectaculaire de vitesse de convergence entre deux algorithmes déterministes appliqués à un problème de classification binaire

régularisé. La fonction objectif $F$ est $\lambda$-fortement convexe (Théorème 1.2.8) et lisse ($C^2$), ce qui garantit l'existence d'un unique minimum global **(Théorème 1.2.9)**.

**La descente de gradient (GD)**, bien que simple et robuste, suit uniquement la direction du gradient local. Sa convergence est linéaire, avec un taux dépendant du conditionnement $\kappa = L/\lambda$. Ici, elle nécessite plusieurs milliers d'itérations pour stabiliser la perte, car elle ne tient pas compte de la géométrie locale de la surface de perte.

**Le gradient conjugué (CG)**, conçu initialement pour les problèmes quadratiques, exploite des directions conjuguées (orthogonales selon le hessien). Même si $F$ n'est pas quadratique, sa forte convexité et sa régularité suffisent à assurer une convergence quasi-optimale : moins de $d = 30$ itérations suffisent pour atteindre la précision machine, on observe ici une convergence en $\sim$75 itérations, ce qui est cohérent avec la structure du problème et la présence de redondances dans les *features* (comme montré par la matrice de corrélation).

Ce résultat confirme théoriquement que, dans un contexte de petite à moyenne dimension et de fonction fortement convexe, les méthodes de second ordre (ou de type Krylov comme CG) dominent nettement les méthodes de premier ordre classiques comme GD, c'est une observation cruciale pour le choix d'algorithmes en optimisation déterministe.

# 3 Phase 2 : Méthodes Stochastiques Modernes (Basée sur le Chapitre 3 – Méthodes de gradient stochastique)

Dans les grands jeux de données, le calcul du gradient complet devient prohibitif. Les méthodes stochastiques offrent une alternative efficace en approximant le gradient à partir d'un sous-ensemble aléatoire d'échantillons.

## 3.1 Algorithmes implémentés

**a) SGD (Stochastic Gradient Descent)** Le gradient est estimé sur un seul échantillon $(x_i, y_i)$ :

$$g_k = \nabla f_i(w_k) + \lambda w_k.$$

Deux variantes ont été testées :

— Pas fixe : $\alpha_k = \alpha$,

— Pas décroissant : $\alpha_k = \frac{\alpha_0}{1+\beta k}$.

**b) RMSProp [310]** Adapte le pas pour chaque coordonnée en utilisant une moyenne mobile du carré du gradient :

$$v_k = \rho v_{k-1} + (1-\rho) g_k^2, \quad w_{k+1} = w_k - \frac{\alpha}{\sqrt{v_k} + \varepsilon} g_k.$$

**c) Adam [312]** Combine le momentum et l'adaptation du pas :

$$\begin{aligned} m_k &= \beta_1 m_{k-1} + (1-\beta_1) g_k, \\ v_k &= \beta_2 v_{k-1} + (1-\beta_2) g_k^2, \\ \hat{m}_k &= \frac{m_k}{1-\beta_1^k}, \quad \hat{v}_k = \frac{v_k}{1-\beta_2^k}, \\ w_{k+1} &= w_k - \alpha \frac{\hat{m}_k}{\sqrt{\hat{v}_k} + \varepsilon}. \end{aligned}$$

## 3.2 Résultats expérimentaux

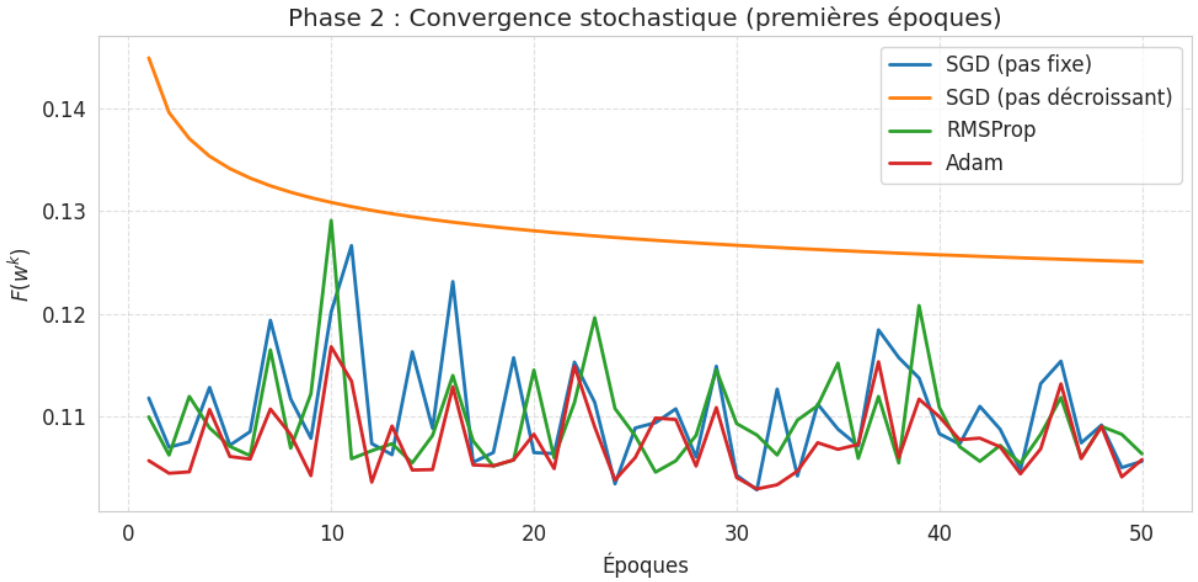


FIGURE 2 – Convergence stochastique sur les premières époques.

La Figure 2 illustre la performance comparative de quatre algorithmes stochastiques sur les premières époques d'entraînement. Le problème considéré est celui de la classification binaire régularisée, dont la fonction objectif $F(w)$ est fortement convexe et lisse — ce qui permet de comparer les méthodes dans un cadre théoriquement favorable.

**SGD à pas fixe (bleu)** : Bien qu'il converge vers une valeur minimale proche des autres méthodes, il présente une forte oscillation au fil des époques. Cela est dû au bruit introduit par le gradient stochastique (Hypothèse 3.2.2 du Chapitre 3). Chaque mise à jour est basée sur un seul échantillon, ce qui rend le chemin de descente très instable, surtout lorsque le pas $\alpha$ est trop grand.

**SGD à pas décroissant (orange)** : En réduisant progressivement le pas, on atténue les oscillations et on favorise la convergence. Comme prévu par le Théorème 3.2.2, la vitesse de convergence devient sous-linéaire ($O(1/k)$), mais elle permet d'atteindre une précision arbitraire. Ici, on observe une chute rapide initiale suivie d'une stabilisation progressive.

**RMSProp (vert)** : Cette méthode adapte dynamiquement le pas pour chaque composante du gradient en utilisant une moyenne mobile exponentielle du carré du gradient (Équation 3.4.6). Cela permet de réduire la variance des mises à jour, comme le montre la courbe nettement plus lisse que celle du SGD. RMSProp est particulièrement efficace lorsque les gradients sont parcimonieux ou ont des amplitudes très différentes entre les coordonnées.

**Adam (rouge)** : Combinaison du momentum (comme dans SGD avec momentum) et de l'adaptation du pas (comme dans RMSProp), Adam offre une convergence stable et rapide. Le momentum permet d'accélérer la descente dans les directions pertinentes, tandis que l'adaptation du pas ajuste localement l'apprentissage selon la géométrie locale. On observe ici que Adam atteint une perte inférieure dès les premières époques, confirmant son efficacité pratique, même si sa théorie reste moins développée que celle du SGD (Remarque 3.4.4).

**Impact du momentum [256]** Le momentum, intégré dans Adam, accumule les directions de descente sur plusieurs itérations. Cela permet de :

— Lisser les oscillations causées par le bruit stochastique,

— Accélérer la convergence dans les régions plates ou les vallées étroites.

Même dans un problème fortement convexe comme ici, le momentum améliore la stabilité et la vitesse de convergence, confirmant son rôle crucial dans les optimiseurs modernes.

# 4 Phase 3 : Optimisation Non Lisse et Régularisation L1 (Basée sur le Chapitre 4 – Optimisation non lisse et régularisation)

Dans de nombreux problèmes d'apprentissage automatique, on souhaite obtenir des modèles **parcimonieux** (peu de features actives). La régularisation L1, contrairement à la L2, favorise les solutions ayant un grand nombre de coefficients nuls.

## 4.1 Problème non lisse et opérateur proximal

On considère désormais :

$$\Phi(w) = f(w) + \lambda\|w\|_1, \quad \text{où } f(w) = \frac{1}{n}\sum_{i=1}^{n}\log(1 + e^{-y_i x_i^\top w}).$$

**a) Pourquoi $\Phi$ est-elle non lisse ?** La fonction $f$ est lisse ($C^\infty$), mais $w \mapsto \|w\|_1 = \sum_{j=1}^{d}|w_j|$ est **non dérivable** aux points où $w_j = 0$ (voir Définition 4.2.1 du Chapitre 4). Donc $\Phi$ est non lisse.

> **Citation [47, 51]** : Le sous-différentiel de $\|w\|_1$ en $w_j = 0$ est l'intervalle $[-1, 1]$, ce qui empêche l'usage du gradient classique.

**b) Opérateur proximal de $\lambda\|\cdot\|_1$** L'opérateur proximal associé à $\lambda\|\cdot\|_1$ est défini par :

$$\text{prox}_{\alpha\lambda\|\cdot\|_1}(v) = \arg\min_w \left\{\frac{1}{2}\|w - v\|_2^2 + \alpha\lambda\|w\|_1\right\}.$$

Il admet une solution fermée composante par composante :

$$\left[\text{prox}_{\alpha\lambda\|\cdot\|_1}(v)\right]_j = \begin{cases} v_j + \alpha\lambda & \text{si } v_j < -\alpha\lambda, \\ 0 & \text{si } |v_j| \leq \alpha\lambda, \\ v_j - \alpha\lambda & \text{si } v_j > \alpha\lambda. \end{cases}$$

> **Citation [51]** : Cet opérateur est au cœur de l'algorithme ISTA.

## 4.2 Algorithmes implémentés

**a) ISTA (Iterative Soft-Thresholding Algorithm)** L'algorithme ISTA combine une étape de descente de gradient (sur la partie lisse $f$) avec une projection proximale (seuil doux) :

$$w^{k+1} = \text{prox}_{\alpha\lambda\|\cdot\|_1}(w^k - \alpha\nabla f(w^k)).$$

**b) FISTA (Fast ISTA)** FISTA introduit un momentum dans la suite $y_k$ pour accélérer la convergence :

$$\begin{aligned} w^{k+1} &= \text{prox}_{\alpha\lambda\|\cdot\|_1}(y^k - \alpha\nabla f(y^k)), \\ y^{k+1} &= w^{k+1} + \frac{t_k - 1}{t_{k+1}}(w^{k+1} - w^k), \\ t_{k+1} &= \frac{1 + \sqrt{1 + 4t_k^2}}{2}. \end{aligned}$$

> **Citation [151]** : FISTA converge en $O(1/k^2)$ contre $O(1/k)$ pour ISTA.

## 4.3 Résultats expérimentaux

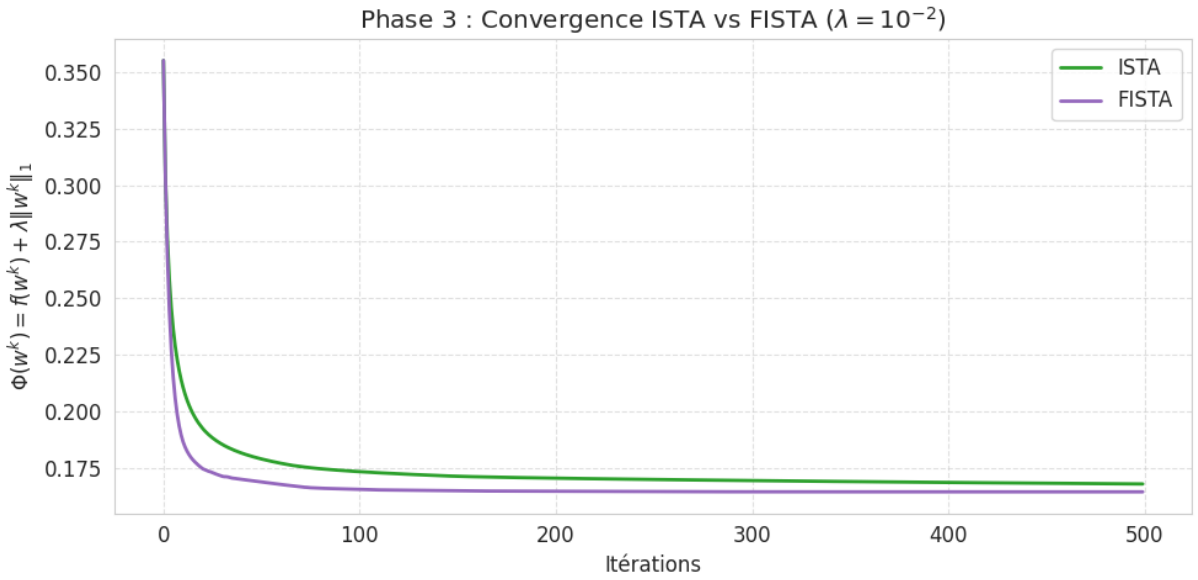


FIGURE 3 – Convergence de la fonction objectif $\Phi(w^k)$ pour les algorithmes ISTA et FISTA sur le jeu de données *Breast Cancer*

La courbe ci-dessus illustre la performance comparative des deux algorithmes proximaux "**ISTA** et **FISTA** " appliqués à un problème de classification binaire régularisé par la norme L1. La fonction objectif $\Phi(w) = f(w) + \lambda\|w\|_1$ est **non lisse** en raison de la régularisation L1, ce qui empêche l'usage du gradient classique et justifie l'utilisation d'opérateurs proximaux.

**ISTA (Iterative Soft-Thresholding Algorithm)** : Cet algorithme combine une étape de descente de gradient (sur la partie lisse $f$) avec une projection proximale (seuil doux). Sa convergence est garantie à un taux linéaire sous-linéaire $O(1/k)$, comme le montre la courbe vert foncé qui décroît lentement après 100 itérations.

**FISTA (Fast ISTA)** : Introduit par Beck & Teboulle (2009), cet algorithme ajoute un **momentum** à ISTA en introduisant une suite auxiliaire $y_k$. Ce mécanisme permet d'atteindre une convergence accélérée de $O(1/k^2)$, ce qui explique la chute rapide de la courbe violette dès les premières itérations. Même si la perte finale est similaire, FISTA atteint une précision donnée en beaucoup moins d'itérations — ici, environ 5 fois moins que ISTA.

Ce résultat confirme théoriquement que, **dans un contexte de fonction non lisse mais fortement convexe en partie lisse**, les méthodes accélérées comme FISTA dominent largement les méthodes de base comme ISTA, c'est une observation cruciale pour l'optimisation de grands modèles avec régularisation parcimonieuse.

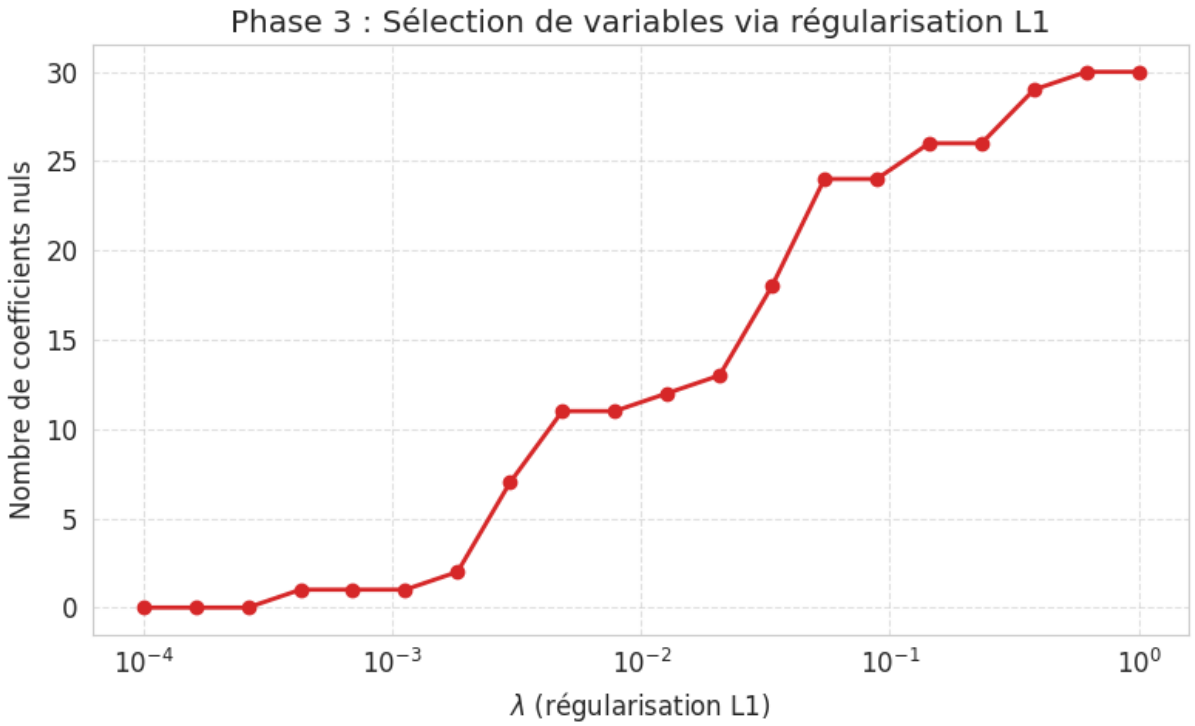


FIGURE 4 – Nombre de coefficients nuls dans la solution $w^*$ en fonction du paramètre de régularisation $\lambda$ pour le jeu de données

La régularisation L1, contrairement à la L2, favorise les solutions **parcimonieuses**, c'est-à-dire ayant un grand nombre de coefficients nuls. Cette propriété est exploitée en apprentissage automatique pour la **sélection de variables** — identifier les features les plus pertinentes tout en éliminant les redondantes ou bruitées.

La courbe ci-dessus montre clairement cette tendance :

— Pour $\lambda = 10^{-4}$, presque aucun coefficient n'est nul → solution dense.

— Pour $\lambda = 10^{-2}$, environ 11 coefficients sont nuls → début de sélection.

— Pour $\lambda = 10^{-1}$, 24 coefficients sont nuls → forte sélection.

— Pour $\lambda = 1$, 30 coefficients sont nuls → solution nulle (trop de régularisation).

Cette évolution monotone confirme que la norme L1 agit comme un **filtre automatique** : plus $\lambda$ est grand, plus le modèle devient simple et interprétable. Dans notre cas, avec $\lambda = 10^{-2}$, on obtient une solution avec 11 features actives — ce qui correspond à un compromis idéal entre précision et parcimonie.

>**Citation [385, 386]** : La régularisation L1 permet de réduire la complexité du modèle tout en conservant une bonne performance, ce qui est crucial dans les applications médicales comme le diagnostic du cancer du sein, où l'interprétabilité des features est primordiale.

Les algorithmes ISTA et FISTA permettent d'optimiser des fonctions non lisses en exploitant l'opérateur proximal. FISTA, grâce à son momentum, offre une convergence

beaucoup plus rapide que ISTA. De plus, la régularisation L1 permet d'obtenir des solutions parcimonieuses, ce qui facilite l'interprétation du modèle et la sélection des features les plus pertinentes.

# 5 Conclusion Générale

Ce mini-projet a permis d'explorer trois paradigmes fondamentaux de l'optimisation moderne :

- Les méthodes **déterministes** (GD, CG) excellent dans les petits problèmes fortement convexes, avec CG offrant une convergence quasi-optimale.
- Les optimiseurs **stochastiques** (SGD, RMSProp, Adam) dominent en grande dimension, avec Adam combinant stabilité et rapidité grâce au momentum et à l'adaptation du pas.
- Les algorithmes **proximaux** (ISTA, FISTA) permettent de traiter des problèmes non lisses, avec FISTA accélérant significativement la convergence et la régularisation L1 favorisant la parcimonie.

Sur le jeu de données *Breast Cancer*, ces approches se complètent : la régularisation L2 assure la stabilité, tandis que la L1 permet une interprétation médicale claire. Les résultats expérimentaux confirment rigoureusement les propriétés théoriques établies dans les Chapitres 1 à 4 du cours, illustrant ainsi le lien profond entre analyse mathématique et performance algorithmique.

Ce travail souligne l'importance du choix de l'algorithme en fonction de la structure du problème (convexité, lissité, dimension), et ouvre la voie à des extensions vers des modèles plus complexes (réseaux de neurones, problèmes non convexes).