



**Nom** : Anass El Moubaraki

**Ecole** : Centrale Nantes

**Formation** : Cycle ingénieur deuxième année

**Option** : Mathématiques et applications, Parcours statistiques et sciences de données

**Stage** : 20 semaines, du 03 avril au 25 août

---

## Forte régularisation de linéarité par morceaux à partir de la norme $l_0$ . Application à l'estimation du coefficient de reproduction lié à la Covid-19

---

**Résumé** : À partir d'un modèle statistique de propagation d'une épidémie nous nous proposons d'estimer le coefficient de reproduction lié à la Covid-19. L'approche régularisée avec une pénalisation  $l_1$  ne permet pas de déterminer "efficacement" la dynamique de la pandémie. Nous adoptons alors une approche régularisée avec une pénalisation  $l_0$ . Cette dernière donne une "meilleure" estimation du coefficient de reproduction et permet ainsi de conclure quant à l'évolution de la dynamique de la pandémie.

**Mots clefs** : épidémiologie, estimation, parcimonie, optimisation convexe

Stage encadré par :

**Mme Barbara Pascal**

Chargée de recherche (LS2N - CNRS/École Centrale de Nantes/Nantes Université).

**Mr Gwenael Samain**

Doctorant (LS2N - CNRS/École Centrale de Nantes/Nantes Université).

**Mr Sébastien Bourguignon**

Maître de conférences (LS2N - CNRS/École Centrale de Nantes/Nantes Université).



## Table des matières

<b>1</b>	<b>Mon expérience au sein de l'organisme d'accueil</b>	<b>4</b>
<b>2</b>	<b>Introduction</b>	<b>6</b>
<b>3</b>	<b>État de l'art</b>	<b>7</b>
3.1	Modèle statistique de propagation d'une épidémie . . . . .	7
3.2	Estimation régularisée du coefficient de reproduction . . . . .	10
3.3	Optimisation convexe non lisse . . . . .	11
3.3.1	Théorie et schéma algorithmique . . . . .	11
3.3.2	Application au problème à l'analyse . . . . .	13
3.4	Résultats numériques . . . . .	15
3.4.1	Vérification de la convergence . . . . .	15
3.4.2	Estimées du coefficient de reproduction . . . . .	17
3.5	Limites de l'approche à l'analyse . . . . .	18
<b>4</b>	<b>Contributions</b>	<b>19</b>
4.1	Parcimonie à la synthèse (pénalisation $l_1$ ) . . . . .	19
4.1.1	Formulation variationnelle . . . . .	20
4.1.2	Schéma algorithmique . . . . .	20
4.1.3	Résultats de simulations . . . . .	22
4.1.4	Limites du modèle d'optimisation . . . . .	27
4.2	Passage à la pénalisation $l_0$ . . . . .	28
4.2.1	Problème à résoudre . . . . .	28
4.2.2	Méthode de résolution . . . . .	28
4.2.3	Résultats de simulations . . . . .	37
4.3	Comparaison entre les résultats fournis dans l'état de l'art et les contributions . . . . .	39
<b>5</b>	<b>Conclusions et perspectives</b>	<b>42</b>
5.1	Conclusions . . . . .	42
<b>6</b>	<b>Annexes</b>	<b>43</b>
6.1	Annexe 1 . . . . .	43
6.2	Annexe 2 . . . . .	43
6.3	Annexe 3 . . . . .	43
<b>7</b>	<b>Bibliographie</b>	<b>44</b>

## **Remerciements**

Je tiens tout d'abord à remercier mes encadrants de stage, Barbara Pascal, Gwenael Samain et Sebastien Bourguignon. Qu'ils soient assurés de toute ma reconnaissance pour m'avoir accordé leur confiance afin d'intégrer leur équipe. Ils m'ont permis, chacun dans son domaine de compétence, d'affiner ma formation scientifique et ce, en me faisant partager, à mon niveau leur goût de la recherche. Les nombreuses discussions que j'ai eu avec eux, leurs précieux conseils, leurs remarques pertinentes, leur souci du détail ainsi que leur disponibilité sans faille m'ont permis de mieux cerner les objectifs de ce stage.

Je remercie Saad Mazen et Françoise Foucher, d'avoir consenti à examiner ce rapport de stage et de me faire l'honneur de participer au jury en vue de m'évaluer.

Je ne saurais oublier les autres membres du laboratoire LS2N. Ils n'ont pas manqué de me manifester leur soutien technique et moral en de maintes occasions.

# 1 Mon expérience au sein de l'organisme d'accueil

Mon stage s'est déroulé au laboratoire des sciences du numérique de Nantes LS2N. Ce dernier a été créé en janvier 2017 afin de répondre à l'ambition et faire progresser la recherche en sciences du numérique à Nantes. Le laboratoire en question met en commun les forces de recherche dans le domaine du numérique de trois établissements d'enseignement supérieur (Université de Nantes, Centrale Nantes, l'IMT Atlantique/campus de Nantes), du CNRS et de son partenaire Inria. Le LS2N est la plus grosse unité de recherche publique sur le site de Nantes et en région Pays de la Loire. Nous présentons la structure du laboratoire à partir de l'organigramme 1.

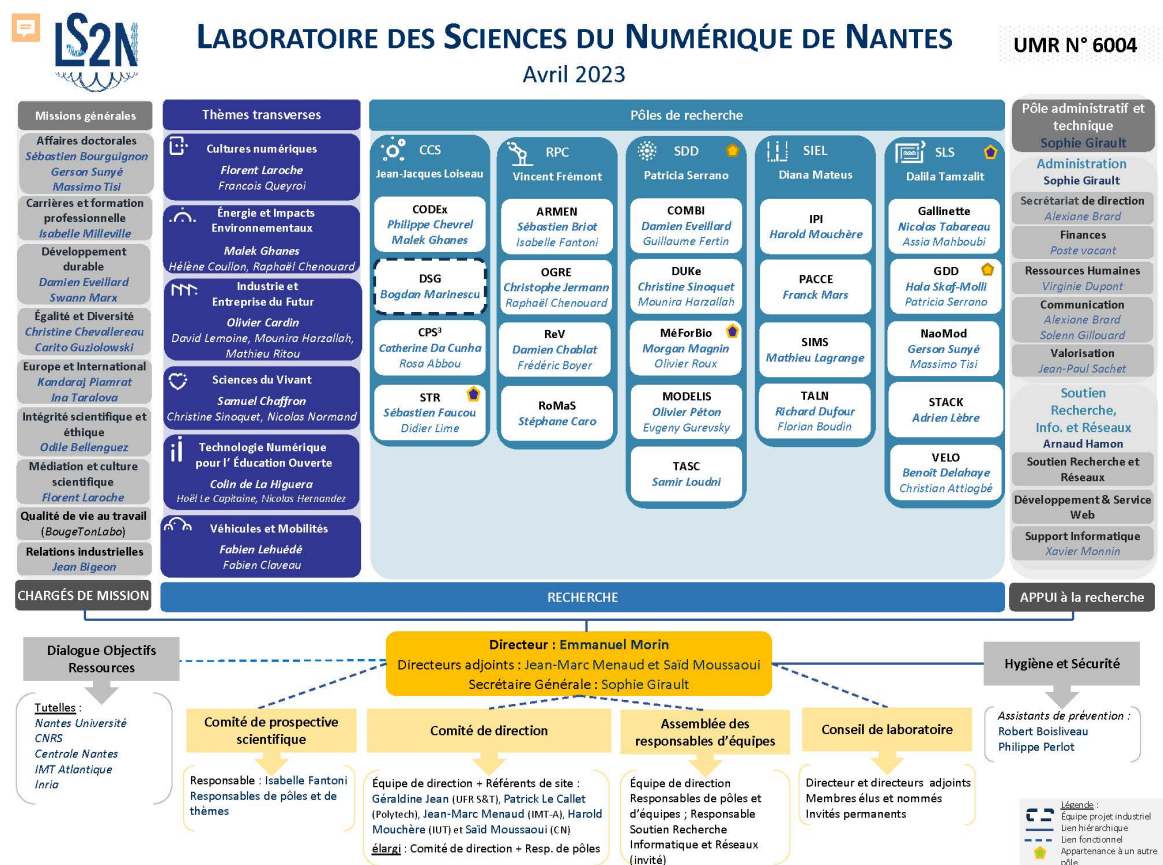


FIGURE 1 – Organigramme du laboratoire LS2N

Pendant mon stage, j'ai eu l'honneur de travailler au sein de l'équipe SIMS (signal, image, son). Avec l'aide de mes encadrants j'ai pu acquérir plusieurs compétences sur les deux plans personnel et professionnel. En effet, pendant cette période j'ai eu l'immense plaisir d'échanger avec différents membres de l'équipe (doctorants stagiaires et chercheurs) ce qui m'a permis de dé-

couvrir le monde de la recherche en général et le travail en équipe sur un projet scientifique en particulier. En termes de compétences techniques, ce stage était l'occasion parfaite pour affûter mes connaissances en programmation latex, en mathématiques théoriques et surtout en analyse convexe, en recherche bibliographique et en analyse de documents scientifiques. J'ai aussi appris à concilier acquis théoriques et programmation, interpréter des résultats de simulations et assimiler les retours critiques de mes encadrants en vue d'améliorer le projet.

En somme, ce stage m'a aidé pour développer mes compétences d'analyse et de synthèse d'articles scientifiques et la manipulation de langages scientifiques dans le but de présenter au mieux un contenu scientifique et d'interagir sans difficulté avec des experts du domaine.

## 2 Introduction

La crise sanitaire liée à la propagation du Covid-19 est l'une des plus importantes crises humanitaires que le monde a connu durant les dernières décennies. En effet, dès son apparition dans la ville de Wuhan en Chine, les pays du monde entier ont mobilisé leurs experts en épidémiologie afin de trouver un vaccin d'une part et pour suivre l'évolution de la dynamique de la pandémie d'autre part. Ce dernier point nécessite une connaissance des paramètres intrinsèques à la pandémie comme le nombres d'infections par exemple mais aussi une bonne maîtrise des outils mathématiques et algorithmiques pour le traitement de données, l'optimisation et l'estimation statistique des paramètres d'intérêt.

Dans le présent travail, nous nous intéresserons à l'estimation régularisée du coefficient de reproduction avec une pénalisation  $l_0$ . Ce paramètre permet de prédire une atténuation exponentielle de la pandémie s'il est inférieur strictement à 1 et une croissance exponentielle s'il est strictement supérieur à 1.

Dans un premier temps, nous ferons l'état de l'art des méthodes de l'estimation régularisée du coefficient de reproduction avec une pénalisation  $l_1$ , puis nous mettrons en lumière les limites de ces modèles et la nécessité d'adopter une approche régularisée avec une pénalisation  $l_0$ . Nous veillerons à présenter avec rigueur le cadre mathématique des problèmes d'optimisation à résoudre, les schémas algorithmiques associés et enfin l'implémentation de ces derniers en python (voir Annexe). Pour clôturer notre travail, nous réaliserons une étude comparative entre état de l'art et contributions pour vérifier la pertinence de nos résultats.

## 3 État de l'art

### 3.1 Modèle statistique de propagation d'une épidémie

Pour notre étude, nous avons comme données le nombre de nouvelles infections quotidiennes en France pendant une durée  $T$  égale à 35 jours. Nous les notons  $\mathbf{Z} = (Z_1, \dots, Z_T)$  où  $Z_t \in \mathbb{N}$  désigne le nombre de nouvelles infections enregistrées au jour  $t$ . Afin de quantifier et de surveiller l'intensité d'une épidémie, le suivi des nouvelles infections n'est pas suffisant.

En effet, s'intéresser au nombre d'infections c'est détecter une vague épidémique après sa survenue. De plus, l'allure de  $\mathbf{Z}$  (voir Figure 2) montre qu'il est bruité à cause d'un manque de données pendant les week-ends.

Par conséquent, s'appuyer uniquement sur les données ne permet pas de déterminer la dynamique de la pandémie.

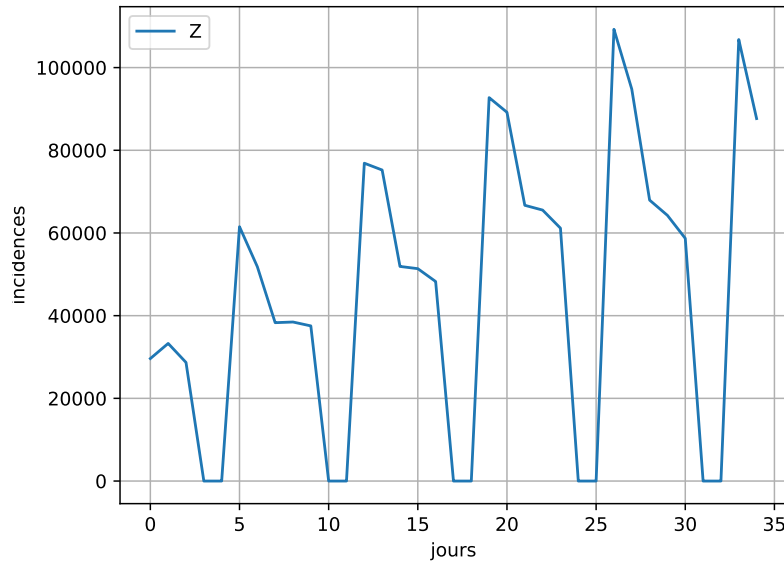


FIGURE 2 – nombres d'infections journalières  $\mathbf{Z} = (Z_1, \dots, Z_{35})$

Pour contourner cette difficulté, nous nous intéresserons à l'estimation du coefficient de reproduction noté  $\mathbf{R}_0$  et défini comme étant le nombre moyen de cas secondaires causés par un seul individu standard atteint du virus pendant sa période de contagion. Dans un premier temps, nous considérons une relaxation temporelle de  $\mathbf{R}_0$  que nous notons  $\mathbf{R}$ . Ce dernier est un vecteur indexé par le temps et dont les composantes  $R_t$  désignent la valeur du coefficient de reproduction au temps  $t$ . Dans ce qui suit nous présentons un modèle

statistique proposé dans [5]. Dans ce modèle, le nombre de nouvelles infections au jour  $t$  suit une loi de poisson dont le paramètre dépend du coefficient de reproduction au jour  $t$  et d'un terme d'inertie prenant en considération les infections aux jours ultérieurs.

Le modèle épidémiologique présenté dans [5] vérifie :

$$\forall t \geq 1 \ (Z_t | Z_1 \dots Z_{t-1}, R_t) \sim \text{Poisson}(R_t \times \Phi_t^Z) \quad (1)$$

$$\Phi_t^Z = \sum_{i=1}^{\min(t, \tau_\phi)} Z_{t-i} \phi_i \quad (2)$$

avec :

- $\tau_\phi$  : durée maximale pour qu'un individu standard reste contagieux.
- $\phi$  : fonction d'intervalle de série également nommée infectiosité et qui représente la distribution des délais entre l'apparition de symptômes dans le cas primaire et le cas secondaire .Cette distribution est tracée dans la Figure 3.

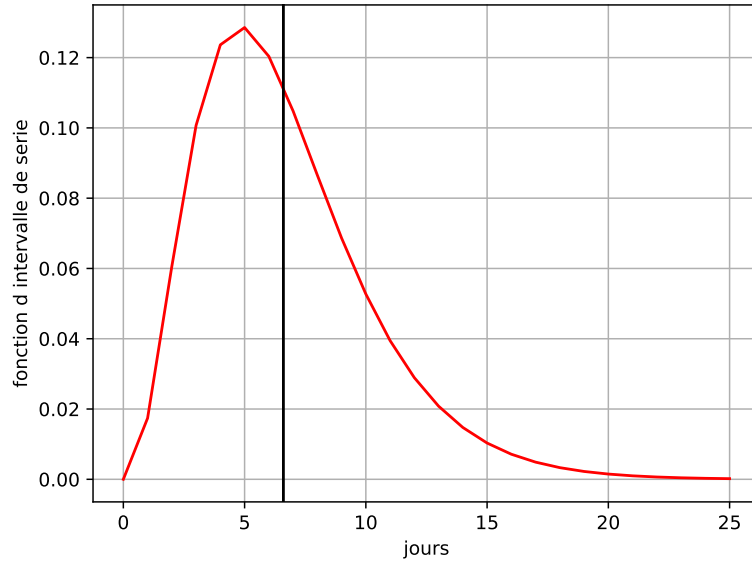


FIGURE 3 – fonction d'intervalle de série

La fonction d'intervalle de série suit une loi  $\Gamma$  (en rouge sur la figure 3) de moyenne 6,6 (en noir sur la figure 3) et d'écart type 3.5. Son allure met en évidence la période où l'individu est très contagieux (pic de la courbe en rouge) qui se situe au niveau du 5 ème jour après l'infection. Nous y voyons aussi la durée maximale de contagion ( $\tau_\phi$ ) qui est égale à 26 jours.



Une première approche pour estimer  $\mathbf{R}$  est de calculer  $\mathbf{R}^{\text{MLE}}$  l'estimateur de maximum de vraisemblance associé au modèle statistique. Pour cela nous cherchons :

$$\begin{aligned}
\mathbf{R}^{\text{MLE}} &\in \underset{\mathbf{R} \in \mathbb{R}^T}{\text{Argmax}} \log(\mathbb{P}(Z_1, \dots, Z_T | R_1, \dots, R_T)) \\
&\Leftrightarrow \mathbf{R}^{\text{MLE}} \in \underset{\mathbf{R} \in \mathbb{R}^T}{\text{Argmax}} \sum_{t=1}^T \log(\mathbb{P}(Z_t | Z_1, \dots, Z_{t-1}, R_t)) \\
&\Leftrightarrow \mathbf{R}^{\text{MLE}} \in \underset{\mathbf{R} \in \mathbb{R}^T}{\text{Argmax}} \sum_{t=1}^T Z_t \log(R_t \Phi_t^Z) - R_t \Phi_t^Z - \log(Z_t!) \\
&\Leftrightarrow \mathbf{R}^{\text{MLE}} \in \underset{\mathbf{R} \in \mathbb{R}^T}{\text{Argmin}} D_{KL}(\mathbf{Z} | \Phi^{\mathbf{Z}} \odot \mathbf{R})
\end{aligned}$$

où  $D_{KL}$  désigne la divergence de Kullback-Leibler et est égale à une constante près à  $-\log(\mathbb{P}(Z_1, \dots, Z_T | R_1, \dots, R_T))$  (voir équation (3) et (4)). La constante ne dépend que des données.

**Expression de la divergence de Kullback-Leibler :**

$$D_{KL}(\mathbf{Z} | \Phi^{\mathbf{Z}} \odot \mathbf{R}) = \sum_{t=1}^T d_{KL}(Z_t | \Phi_t^Z \odot R_t) \quad (3)$$

avec :

$$d_{KL}(Z_t | \Phi_t^Z \odot R_t) = \begin{cases} Z_t \log(\frac{Z_t}{R_t \Phi_t^Z}) + R_t \Phi_t^Z - Z_t & \text{si } Z_t > 0 \text{ et } R_t > 0 \\ R_t \Phi_t^Z & \text{si } Z_t = 0 \text{ et } R_t \geq 0 \\ +\infty & \text{sinon} \end{cases} \quad (4)$$

Le minimum est obtenu en :

$$\mathbf{R}^{\text{MLE}} = \left( \frac{Z_t}{\Phi_t^Z} \right)_{1 \leq t \leq T} \quad (5)$$

Nous traçons  $\mathbf{R}^{\text{MLE}}$  dans la figure 4 :

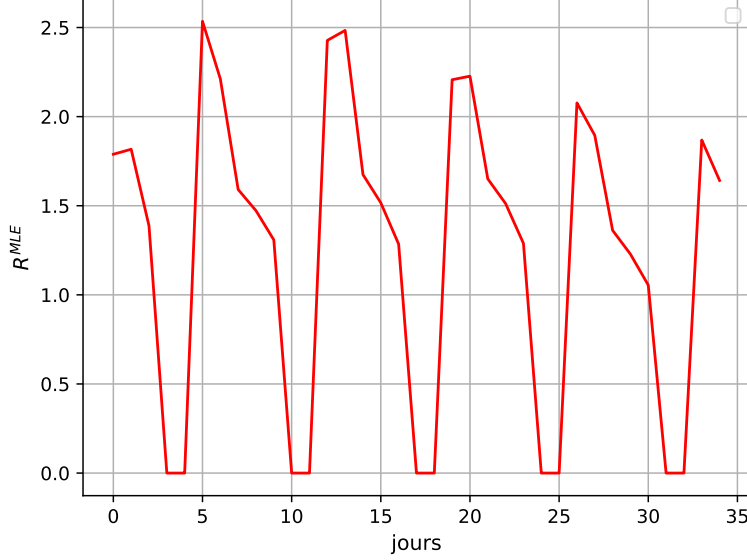


FIGURE 4 – Estimation du coefficient de reproduction par maximum de vraisemblance

L'analyse de figure 4 montre que  $\mathbf{R}^{MLE}$  est bruité et ne permet pas de déterminer la dynamique de la pandémie. En effet, l'estimateur obtenu a une allure similaire à celle de  $\mathbf{Z}$  et ne permet pas de conclure.

Il est donc nécessaire de considérer une approche régularisante qui nous donnera un estimateur de  $\mathbf{R}$  fidèle au modèle épidémiologique continu dans le temps et non bruité.

### 3.2 Estimation régularisée du coefficient de reproduction

Dans cette partie nous nous intéresserons à l'estimation régularisée de  $\mathbf{R}$  avec une pénalisation  $l_1$  son Laplacien [1].

Nous cherchons alors à résoudre :

$$\min_{\mathbf{R} \in \mathbb{R}^T} D_{KL}(\mathbf{Z} | \Phi^{\mathbf{Z}} \odot \mathbf{R}) + \lambda \|\mathbf{D}_2 \mathbf{R}\|_1 \quad (6)$$

où :

- $\lambda$  désigne le coefficient de régularisation.
- $\mathbf{D}_2 \in \mathcal{M}_{T-2 \times T}(\mathbb{R})$  désigne l'opérateur Laplacien discret. Son expression se trouve dans (7) et un exemple pour des données de taille  $T = 5$  est fourni dans (8).

$$(\mathbf{D}_2)_{i,j} = \begin{cases} 1 & \text{si } j = i \text{ et } j = i + 2 \\ -2 & \text{si } j = i + 1 \\ 0 & \text{sinon} \end{cases} \quad (7)$$

Expression de  $\mathbf{D}_2$  pour  $T = 5$  :

$$\begin{pmatrix} 1 & -2 & 1 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 1 & -2 & 1 \end{pmatrix} \quad (8)$$

Remarquons que  $\|\cdot\|_1$  est convexe et non différentiable et  $D_{KL}$  est convexe non différentiable et à valeur dans  $\mathbb{R} \cup \{\infty\}$ . Par conséquent, la résolution du problème d'optimisation (6) nécessite des outils d'optimisation convexe non lisse que nous détaillerons dans ce qui suit.

### 3.3 Optimisation convexe non lisse

#### 3.3.1 Théorie et schéma algorithmique

Soient  $\mathcal{X}, \mathcal{Y}$  deux espaces de Hilbert de dimension finie.

Soient  $F : \mathcal{Y} \rightarrow \mathbb{R} \cup \{\infty\}$  et  $G : \mathcal{X} \rightarrow \mathbb{R} \cup \{\infty\}$  deux fonctions propres, convexes, semi-continues inférieurement et  $\mathbf{K} : \mathcal{X} \rightarrow \mathcal{Y}$  un opérateur linéaire continu. (H.1)

Nous considérons le problème de minimisation suivant :

$$\min_{x \in \mathcal{X}} F(\mathbf{K}x) + G(x) \quad (9)$$

Nous supposons de plus que le problème (9) admet une solution (H.2).

La résolution du problème ci-dessous relève plusieurs difficultés. En effet, les fonctions  $F$  et  $G$  ne sont potentiellement pas différentiables. Par conséquent, nous ne pouvons pas appliquer des algorithmes de type descente de gradient de pas  $\tau > 0$  (voir équation (10)). Une solution serait donc d'utiliser un algorithme du point proximal (voir équation (11)).

$$x^{[k+1]} = x^{[k]} + \tau \nabla (F(\mathbf{K}.) + G)(x^{[k]}) \quad (10)$$

$$x^{[k+1]} = \text{prox}_{\tau(F(\mathbf{K}.)+G)}(x^{[k]}) \quad (11)$$

où  $\text{prox}$  désigne l'opérateur proximal et vérifie :

$$\text{prox}_{\tau(F(\mathbf{K}.)+G)}(x^{[k]}) = \underset{z \in \mathcal{X}}{\text{Argmin}} \tau(F(\mathbf{K}.) + G)(z) + \frac{\|z - x^{[k]}\|_2^2}{2}$$

Nous montrons dans la proposition 1 que l'algorithme du point proximal est un algorithme du point fixe qui fait intervenir la notion de sous-différentielle

(que nous définissons dans l'équation 12) en tant que généralisation du gradient pour les fonctions non différentiables.

Soit  $f : \mathcal{X} \rightarrow \mathbb{R} \cup \{\infty\}$  une fonction convexe.

$$\partial f(x) = \{s \in \mathcal{X} \text{ tq } \forall y \in \mathcal{X} \ f(y) - f(x) \geq \langle s, y - x \rangle\} \quad (12)$$

**Proposition 1.** *Soit  $f : \mathcal{X} \rightarrow \mathbb{R} \cup \{\infty\}$  une fonction propre, convexe et semi-continue inférieurement alors :*

$$\text{prox}_{\tau f}(y) = (I + \tau \partial f)^{-1}(y) \quad (13)$$

*Démonstration.*  $(I + \tau \partial f) : \mathcal{X} \rightarrow 2^{\mathcal{X}}$  :

$$\forall x \in \mathcal{X} : (I + \tau \partial f)(x) = \{x\} + \tau \partial f(x) = \{x + \tau s \text{ tq } s \in \partial f(x)\}$$

$\forall y \in \mathcal{X} : p_y = \text{prox}_{\tau f}(y)$  existe [4].

$$\begin{aligned} 0 \in \partial\left(\frac{\|y - \cdot\|_2^2}{2\tau}\right)(p_y) + \partial f(p_y) &\Leftrightarrow 0 \in \left\{\frac{p_y - y}{\tau}\right\} + \partial f(p_y) \\ &\Leftrightarrow y \in (I + \tau \partial f)(p_y) \\ &\Leftrightarrow p_y \in (I + \tau \partial f)^{-1}(y) \\ &\Leftrightarrow p_y = (I + \tau \partial f)^{-1}(y) \end{aligned}$$

$\partial\left(\frac{\|y - \cdot\|_2^2}{2\tau}\right)(p_y) = \left\{\frac{p_y - y}{\tau}\right\}$  car  $\frac{\|y - \cdot\|_2^2}{2\tau}$  est convexe de classe  $C^1$  et donc la sous-différentielle en un point  $x$  se réduit au singleton gradient en ce même point.  $\square$

Le calcul de l'opérateur proximal associé à  $F(K \cdot) + G$  est très complexe à cause de la présence d'une composée d'une fonction et d'un opérateur linéaire à priori non trivial. Pour remédier à ce problème, nous nous intéresserons à la recherche d'un point selle du problème primal-dual de Fenchel associé au problème général (14).

$$\min_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} \langle \mathbf{K}x, y \rangle + G(x) - F^*(y) \quad (14)$$

ou,  $F^*$  désigne la fonction conjuguée de  $F$ . Elle vérifie :

$$(\forall y \in \mathcal{Y}) \ F^*(y) = \max_{z \in \mathcal{Z}} \langle z, y \rangle - F(z) \quad (15)$$

**Proposition 2** (Équivalence entre la résolution du problème général et la recherche du point selle du primal dual). *Soit  $F, G$  et  $\mathbf{K}$  vérifiant (H.1)*

$$\min_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} \langle \mathbf{K}x, y \rangle + G(x) - F^*(y) = \min_{x \in \mathcal{X}} F(\mathbf{K}x) + G(x)$$

*Démonstration.* La preuve repose sur l'identité de Fenchel-Moreau [4]  
En effet, si  $F$  est une fonction propre, convexe et semi-continue inférieurement alors :

$$F^{**} = F$$

Ainsi :

$$\forall y \in \mathcal{Y} : F(y) = F^{**}(y) = \max_{z \in \mathcal{Y}} \langle z, y \rangle - F^*(z)$$

Par conséquent :

$$\begin{aligned} \min_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} \langle \mathbf{K}x, y \rangle + G(x) - F^*(y) &= \min_{x \in \mathcal{X}} G(x) + \max_{y \in \mathcal{Y}} \langle \mathbf{K}x, y \rangle - F^*(y) \\ &= \min_{x \in \mathcal{X}} G(x) + F^{**}(\mathbf{K}x) \\ &= \min_{x \in \mathcal{X}} G(x) + F(Kx) \end{aligned}$$

□

L'algorithme 1 est fourni dans [3]. Il permet d'obtenir une estimée du point selle du problème primal-dual (14).

---

**Algorithm 1** Schéma algorithmique [3]

---

**Entrées:**  $F, G, \mathbf{K}$

- 1: Paramètres :  $k_f, \theta, \tau, \sigma$
- 2: **for**  $k = 1, \dots, k_f$  **do**
- 3:  $y^{[k+1]} = y^{[k]} + \sigma \mathbf{K}x^{[k]} - \sigma \text{prox}_{\frac{1}{\sigma}F}(\frac{1}{\sigma}y^{[k]} + \mathbf{K}x^{[k]})$
- 4:  $x^{[k+1]} = \text{prox}_{\tau G}(x^{[k]} - \tau \mathbf{K}^* \bar{y}^{[k+1]})$
- 5:  $\bar{y}^{[k+1]} = y^{[k+1]} + \theta(y^{[k+1]} - y^{[k]})$
- 6: **end for**

**Sorties:**  $(\hat{x}, \hat{y}) \leftarrow (x^{k_f}, y^{k_f})$

---

**Théorème 1** (Convergence de l'algorithme de Chambolle-Pock). [3]

Soient  $F, G$ , et  $\mathbf{K}$  vérifiant (H.1) et (H.2).

On suppose que  $\|\mathbf{K}\|_2 = L$ ,  $\theta = 1$  et  $\tau\sigma L^2 < 1$  alors :

L'algorithme 1 converge vers un point selle du problème primal-dual.

### 3.3.2 Application au problème a l'analyse

Nous rappelons que nous cherchons :

$$\mathbf{R}^\dagger \in \underset{\mathbf{R} \in \mathbb{R}^T}{\text{Argmin}} D_{KL}(\mathbf{Z} | \Phi^{\mathbf{Z}} \odot \mathbf{R}) + \|\lambda \mathbf{D}_2 \mathbf{R}\|_1$$

Les fonctions  $D_{KL}$  et  $\|\cdot\|_1$  sont convexes, propres, semi-continues inférieurement et  $\lambda \mathbf{D}_2$  est un opérateur linéaire continu. De plus, le problème (6) s'écrit sous la forme  $\min_{x \in \mathcal{X}} F(\mathbf{K}x) + G(x)$ . La Table 1 identifie les composantes du problème général d'optimisation convexe non lisse a celles du problème 6.

problème général	Estimation régularisée avec pénalisation sur le Laplacien
$\mathcal{X}$	$\mathbb{R}^T$
$\mathcal{Y}$	$\mathbb{R}^{T-2}$
$x$	$\mathbf{R}$
$F$	$\ \cdot\ _1$
$G$	$D_{KL}(\mathbf{Z} \Phi^{\mathbf{Z}} * .)$
$\mathbf{K}$	$\lambda \mathbf{D}_2$

TABLE 1 – Identification des composantes du problème de l’estimation régularisée a celles du problème général

**Théorème 2** (Vérification de l’hypothèse (H.2) pour le problème 6). [8]  
*Soient  $T > 0$ ,  $\mathbf{Z} \in (\mathbb{R}^+)^T$  et  $\Phi^{\mathbf{Z}} \in (\mathbb{R}^{*+})^T$  :*

$$\exists \mathbf{R}^\dagger \in \underset{\mathbf{R} \in \mathbb{R}^T}{\text{Argmin}} D_{KL}(\mathbf{Z}|\Phi^{\mathbf{Z}} \odot \mathbf{R}) + \|\lambda \mathbf{D}_2 \mathbf{R}\|_1$$

Afin d’implémenter l’algorithme de Chambolle-Pock et résoudre le problème (6), nous donnons les expressions des opérateurs proximaux associés à  $\|\cdot\|_1$  et  $D_{KL}(\mathbf{Z}|\Phi^{\mathbf{Z}} \odot .)$  ([1]).

**Proposition 3** (Opérateur proximal associe a  $D_{KL}(\mathbf{Z}|\Phi^{\mathbf{Z}} \odot .)$ ). *Soient  $T > 3$ ,  $x \in \mathbb{R}^T$  et  $r > 0$  :*

$$\text{prox}_{rD_{KL}(\mathbf{Z}|\Phi^{\mathbf{Z}} \odot .)}(x) = \left( \frac{x_t - r\Phi_t + \sqrt{|x_t - r\Phi_t|^2 + 4\tau Z_t}}{2} \right)_{1 \leq t \leq T} \quad (16)$$

**Proposition 4** (Opérateur proximal associé à  $\|\cdot\|_1$ ). *Soient  $T > 3$ ,  $y \in \mathbb{R}^{T-2}$  et  $r > 0$  :*

$$\text{prox}_{r\|\cdot\|_1}(y) = (\max(0, |y_t| - r) \text{sign}(y_t))_{1 \leq t \leq T-2} = \left( \max \left( 0, 1 - \frac{r}{|y_t|} \right) y_t \right)_{1 \leq t \leq T-2} \quad (17)$$

Nous présentons dans l’algorithme 2 un schéma qui reflète l’implémentation sur machine de l’algorithme de Chambolle-Pock pour la résolution du problème (6). Le critère d’arrêt utilisé que nous notons  $(\mathcal{A})$  est fourni dans [6]. Il permet de vérifier la convergence via l’évolution des accroissements calculés par rapport aux itérés du coefficient de reproduction  $\mathbf{R}$  et lissés sur une plage de 500 itérations.

---

**Algorithm 2** Schéma algorithmique pour l'estimation régularisée de  $\mathbf{R}$  avec pénalisation sur son Laplacien

---

**Entrées:**  $\mathbf{Z}, \Phi^{\mathbf{Z}}, \mathbf{D}_2$

```

1: Parametres :  $\theta, \lambda, \tau, \sigma, \epsilon, k_{\max}$ 
2:  $k \leftarrow 0$ 
3:  $\mathbf{R}^{[0]} \leftarrow (\mathbf{1})_{1 \leq t \leq T}^\top$ 
4:  $\mathbf{Y}^{[0]} \leftarrow \lambda \mathbf{D}_2 \mathbf{R}^{[0]}$ 
5:  $\mathcal{A}(0) \leftarrow 1 + \epsilon$ 
6: while  $\mathcal{A}(k) > \epsilon$  et  $k < k_{\max}$  do
7:    $\mathcal{O}(k) \leftarrow D_{KL}(\mathbf{Z} | \Phi^{\mathbf{Z}} \odot \mathbf{R}^{[k]}) + \|\lambda \mathbf{D}_2 \mathbf{R}^{[k]}\|_1$ 
8:    $\mathbf{Y}^{[k+1]} \leftarrow \mathbf{Y}^{[k]} + \sigma \lambda \mathbf{D}_2 \mathbf{R}^{[k]} - \sigma \text{prox}_{\frac{1}{\sigma} \|\cdot\|_1}(\frac{1}{\sigma} \mathbf{Y}^{[k]} + \lambda \mathbf{D}_2 \mathbf{R}^{[k]})$ 
9:    $\mathbf{R}^{[k+1]} \leftarrow \text{prox}_{\tau D_{KL}(\mathbf{Z} | \Phi^{\mathbf{Z}} \cdot)}(\mathbf{R}^{[k]} - \tau \lambda \mathbf{D}_2^\top (2 \mathbf{Y}^{[k+1]} - \mathbf{Y}^{[k]}))$ 
10:   $\mathcal{A}(k+1) \leftarrow \max_{\max(0, k-500) < k' \leq k} \max_{1 \leq t \leq T} \frac{|R_t^{[k'+1]} - R_t^{[k']}|}{R_t^{[k']}}$ 
11:   $k \leftarrow k + 1$ 
12: end while
13:  $k_f \leftarrow k$ 
Sorties:  $k_f, \hat{\mathbf{R}} \leftarrow \mathbf{R}^{[k_f]}, \mathcal{A}, \mathcal{O}$ 

```

---

### 3.4 Résultats numériques

**Données :**

Pour les simulations nous utilisons les données présentées dans la figure 2 et que nous normaliserons pour accélérer la convergence. En pratique nous travaillerons avec  $\frac{\mathbf{Z}}{\text{std}(\mathbf{Z})}$  et  $\frac{\Phi^{\mathbf{Z}}}{\text{std}(\mathbf{Z})}$ .

**Choix des paramètres :**

Pour assurer la convergence de l'algorithme de Chambolle-Pock, nous choisissons  $\tau = \sigma = \frac{0.99}{\|\lambda \mathbf{D}_2\|_2}$  et  $\theta = 1$ .

Nous fixons ensuite  $\epsilon = 10^{-8}$  et  $k_{\max} = 150000$  et nous explorons quelques niveaux de régularisation en choisissant  $\lambda \times \text{std}(\mathbf{Z})$  dans  $\{16178, 21371, 49262, 85961, 150000\}$ . Par abus de langage, nous assimilerons dans ce qui suit  $\lambda$  à  $\lambda \times \text{std}(\mathbf{Z})$ .

#### 3.4.1 Vérification de la convergence

Pour vérifier la convergence, nous visualisons pour chaque valeur de  $\lambda$  explorée l'évolution des accroissements ( $\mathcal{A}$ ) en fonction du nombre d'itérations ( $k$ ) (voir figure 5). Nous traçons aussi les itérés de la fonction objectif ( $\mathcal{O}(k)$ ) dans la figure 6.

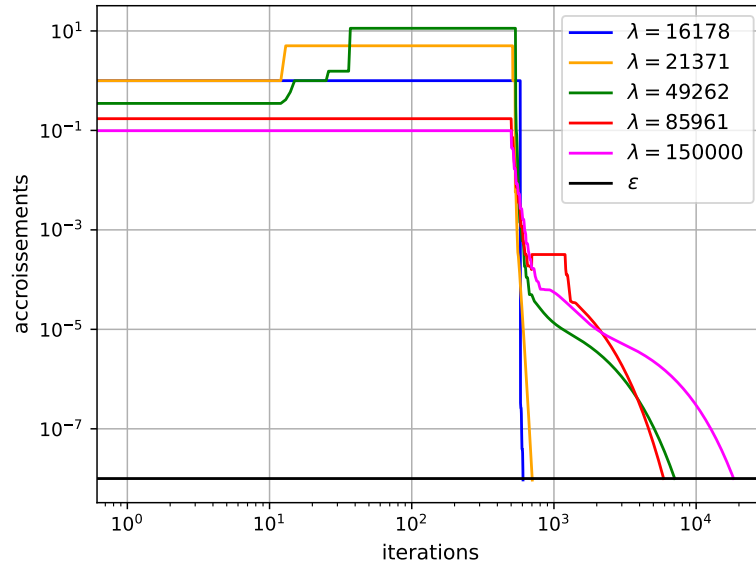


FIGURE 5 – Évolution des accroissements pour quelques valeurs de  $\lambda$

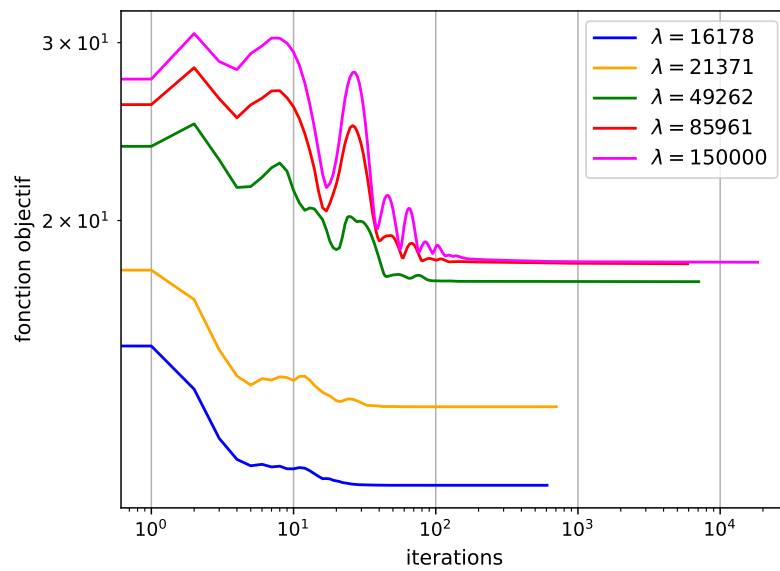


FIGURE 6 – Évolution de la fonction objectif pour quelques valeurs de  $\lambda$

**Commentaire :**



L'analyse des courbes met en évidence une stabilité des accroissements ( pas de fluctuations) pour toutes les valeurs de  $\lambda$  explorées. La fonction  $\mathcal{A}$  décroît progressivement et  $\mathcal{A}(k_f)$  est très proche du seuil d'itération  $\epsilon$ . De plus, la fonction objectif décroît pour atteindre un plateau de saturation reflétant la convergence. La Table 2 récapitule quelques indicateurs de convergence de pour les niveaux de régularisation explorés.

$\lambda$	$\mathcal{A}(k_f)$	$k_f$	$\mathcal{O}(k_f)$
16178	$9.38 \times 10^{-9}$	605	10.93704
21371	$9.31 \times 10^{-9}$	703	13.08159
49262	$9.99 \times 10^{-9}$	7065	17.39835
85961	$9.98 \times 10^{-9}$	5906	18.12375
150000	$9.99 \times 10^{-9}$	18335	18.18421

TABLE 2 – Tableau récapitulatif de quelques indicateurs de convergence

### 3.4.2 Estimées du coefficient de reproduction

Nous traçons dans la figure 7 les estimées de  $\mathbf{R}$  pour visualiser l'effet de la régularisation.

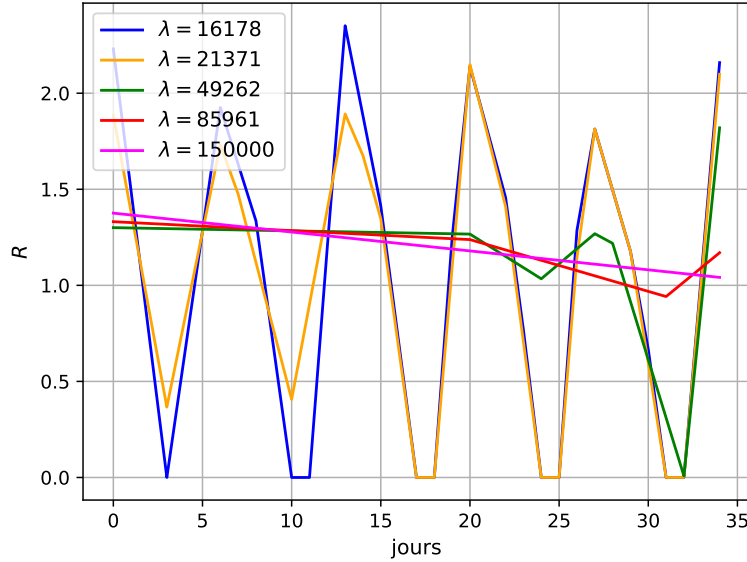


FIGURE 7 – Estimées du coefficient de reproduction

#### Commentaire :

L'analyse des estimées de  $\mathbf{R}$  montre que plus  $\lambda$  augmente, plus le nombre de ruptures de pentes diminue et nous observons une élimination progressif du

bruit (des jours les plus anciens aux jours les plus récents). Cette progression est principalement due à l'amplitude de  $\mathbf{Z}$  avant chaque week-end (voir Figure 2). En effet, plus nous évoluons dans le temps plus l'amplitude est importante et par conséquent les week-ends sont plus difficiles à corriger.

### 3.5 Limites de l'approche à l'analyse

Pour détecter les changements de dynamique de la pandémie nous avons besoin de changement de pentes effectifs et donc de la parcimonie exacte sur les estimées de  $\mathbf{D}_2\mathbf{R}$ . Pour  $\lambda \in \{85961, 150000\}$ , nous observons respectivement 1 seule rupture de pentes et 0 ruptures de pentes pour  $\hat{\mathbf{R}}$ . Cependant, la figure 8 montre que les estimées du Laplacien du coefficient de reproduction ont des composantes dont l'amplitude est non nulle et ce aux instants où nous visualisons pas de changement brusque de dynamique. Par conséquent, nous n'arrivons pas à conclure avec précision par rapport à l'évolution de la pandémie.

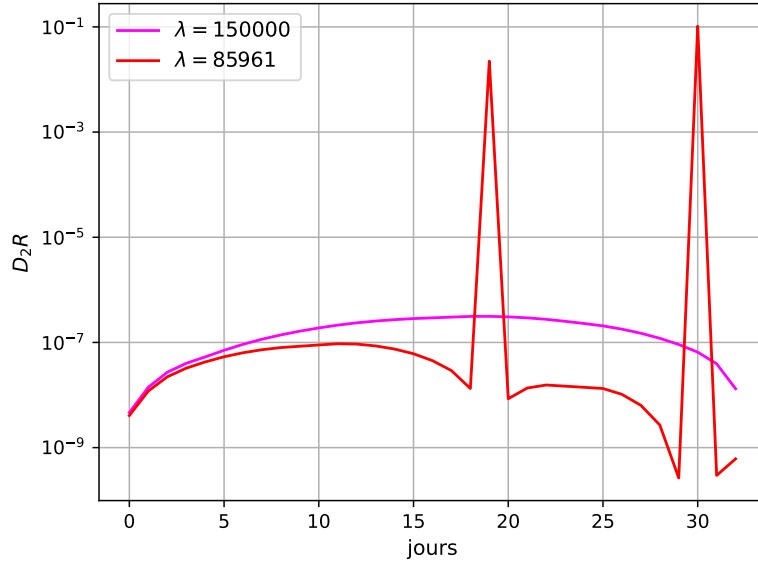


FIGURE 8 – Amplitude des estimées du Laplacien du coefficient de reproduction

Une manière de corriger cette anomalie est d'obtenir de la parcimonie exacte en choisissant une pénalisation plus sévère que la norme  $l_1$  à savoir la norme  $l_0$ . Cette dernière vérifie :

$$\forall x \in \mathbb{R}^n : \|x\|_0 = \text{Card}\{i \in \llbracket 1, n \rrbracket \text{ tel que } x_i \neq 0\}$$

Le problème 6 s'écrit alors comme suit :

$$\min_{\mathbf{R} \in \mathbb{R}^T} D_{KL}(\mathbf{Z}|\Phi^{\mathbf{Z}} \odot \mathbf{R}) + \lambda \|\mathbf{D}_2 \mathbf{R}\|_0 \quad (18)$$

La résolution de 18 pose plusieurs difficultés. D'une part, la norme  $l_0$  n'est pas convexe, d'autre part l'utilisation d'une approche combinatoire sera freinée par la présence de  $\mathbf{D}_2$  qui est un opérateur linéaire non trivial et non inversible. Ces difficultés motivent les contributions apportées dans ce stage.

Dans un premier temps, nous présenterons un problème d'optimisation permettant d'obtenir de la parcimonie sur le vecteur des pentes associé à  $\mathbf{R}$  en utilisant une pénalisation  $l_1$  (parcimonie à la synthèse). Dans un second temps, nous adopterons une approche régularisée avec une pénalisation  $l_0$  en vue de remédier au biais de régularisation introduit par la norme 1 d'une part et pour obtenir de la parcimonie exacte sur le vecteur des pentes d'autre part.

## 4 Contributions

### 4.1 Parcimonie à la synthèse (pénalisation $l_1$ )

Toute fonction linéaire par morceaux ayant un nombre fini de ruptures de pentes peut s'écrire à une constante près comme somme de rampes qui s'activent aux instants où la fonction change de dynamique (voir équation 19 pour la définition d'une rampe). Cela dit, pour obtenir de la parcimonie sur le vecteur des pentes associé à  $\mathbf{R}$  nous écrivons ce dernier sous la forme d'un produit matriciel entre un dictionnaire que nous écrivons par bloc sous forme  $(\mathbf{L}, \mathbf{M})$  et le vecteur des pentes que nous notons  $(\mathbf{u}, \mathbf{v})$  (voir équation 20) :

$$ramp(t) = t \times \mathbb{1}_{t \geq 0} \quad (19)$$

$$\mathbf{R} = \mathbf{L}\mathbf{u} + \mathbf{M}\mathbf{v} \quad (20)$$

où  $\mathbf{L} \in \mathcal{M}_{T,2}(\mathbb{R})$ ,  $\mathbf{M} \in \mathcal{M}_{T,T-2}(\mathbb{R})$ ,  $\mathbf{u} \in \mathbb{R}^2$  et  $\mathbf{v} \in \mathbb{R}^{T-2}$ .

#### Illustration :

Pour  $T = 4$  les composantes du vecteur  $\mathbf{R}$  vérifient :

$$R_t = u_1 + u_2 ramp(t-1) + v_1 ramp(t-2) + v_2 ramp(t-3)$$

En nous appuyant sur la définition d'une rampe et en évaluant à chaque instant  $t \in \llbracket 1, 4 \rrbracket$  nous obtenons :

$$\mathbf{R} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 2 & 1 & 0 \\ 1 & 3 & 2 & 1 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \mathbf{L}\mathbf{u} + \mathbf{M}\mathbf{v}$$

#### 4.1.1 Formulation variationnelle

L'exemple précédent montre que les colonnes du dictionnaire ont des énergies (norme de chaque colonne) différentes. Dans la définition de notre problème, (voir équation (21)) nous considérons un dictionnaire dont les colonnes seront normalisées (voir expressions (22) et (23)) pour homogénéiser la régularisation. De plus, nous ne pénaliserons pas l'offset et la première pente pour laisser un degré de liberté qui déterminera l'allure de  $\hat{\mathbf{R}}$ .

$$\min_{(\mathbf{u}, \mathbf{v}) \in \mathbb{R}^2 \times \mathbb{R}^{T-2}} D_{KL}(\mathbf{Z} | \Phi^{\mathbf{Z}} \odot (\mathbf{L}\mathbf{u} + \mathbf{M}\mathbf{v})) + \lambda \|\mathbf{v}\|_1 \quad (21)$$

où :

- $\mathbf{u} \in \mathbb{R}^2$  : vecteur contenant l'offset et la première pente.
- $\mathbf{v} \in \mathbb{R}^{T-2}$  : vecteur contenant les pentes associées aux rampes qui s'activent à  $2 \leq t \leq T$ .
- $\mathbf{L} \in \mathcal{M}_{T,2}(\mathbb{R})$  : colonnes normalisées associées à  $\mathbf{u}$  et est définie par :

$$L_{i,j} = \begin{cases} \frac{i-1}{\sum_{i=1}^T |i-1|} & \text{si } j = 1 \\ \frac{1}{|T|} & \text{si } j = 2 \\ 0 & \text{sinon} \end{cases} \quad (22)$$

- $\mathbf{M} \in \mathcal{M}_{T,T-2}(\mathbb{R})$  : colonnes normalisées associées à  $\mathbf{v}$  et est définie par :

$$M_{i,j} = \begin{cases} \frac{i-j+1}{\sum_{i=j}^T |i-j+1|} & \text{si } i - j + 1 \geq 0 \\ 0 & \text{sinon} \end{cases} \quad (23)$$

#### 4.1.2 Schéma algorithmique

Les fonctions  $D_{KL}$  et  $(\mathbf{u}, \mathbf{v}) \rightarrow \lambda \|\mathbf{v}\|_1$  sont convexes, propres, semi-continues inférieurement et  $(\mathbf{L}, \mathbf{M})$  est un opérateur linéaire continu. De plus, le problème (21) s'écrit sous la forme  $\min_{x \in \mathcal{X}} F(\mathbf{K}x) + G(x)$ . La Table 3 identifie les composantes du problème général d'optimisation convexe non lisse à celles du problème (21).

problème général	Estimation régularisée avec pénalisation sur le vecteur des pentes
$\mathcal{X}$	$\mathbb{R}^T$
$\mathcal{Y}$	$\mathbb{R}^T$
$x$	$(\mathbf{u}, \mathbf{v})$
$F$	$D_{\text{KL}}(\mathbf{Z}   \Phi^{\mathbf{Z}} \odot .)$
$G$	$(\mathbf{u}, \mathbf{v}) \rightarrow \lambda \ \mathbf{v}\ _1$
$\mathbf{K}$	$(\mathbf{L}, \mathbf{M})$

TABLE 3 – Identification des composantes du problème à la synthèse à celles du problème général

**Théorème 3** (Vérification de l'hypothèse (H.2) pour le problème 21). *Soit  $n \in \mathbb{N}^*$ .*

*Soit  $F : \mathbb{R}^T \rightarrow \mathbb{R} \cup \{\infty\}$  une fonction propre, convexe semi-continue inférieurement et coercive.*

*Soit  $G : \mathbb{R}^T \rightarrow \mathbb{R} \cup \{\infty\}$  convexe, propre, semi-continue inférieurement et bornée inférieurement.*

*Soit  $\mathbf{K} \in \mathcal{M}_{T,n}(\mathbb{R})$  une matrice de rang plein. alors :*

$$(\exists x^\dagger \in \mathbb{R}^T) \text{ tq : } x^\dagger \in \underset{x \in \mathbb{R}^T}{\text{Argmin}} F(\mathbf{K}x) + G(x)$$

*Démonstration.*  $\mathbf{K}$  est une matrice de rang plein et  $F$  coercive alors :  
 $F(\mathbf{K}.)$  est coercive.

$G$  est bornée inférieurement , par conséquent :  
 $F(\mathbf{K}.) + G$  est coercive et semi-continue inférieurement.

Nous appliquons alors le corollaire 11.15 du livre [4] :

$$F(\mathbf{K}.) + G \text{ coercive et semi-continue inférieurement} \Rightarrow (\exists x^\dagger \in \mathbb{R}^T) \text{ tq : } x^\dagger \in \underset{x \in \mathbb{R}^T}{\text{Argmin}} F(\mathbf{K}x) + G(x)$$

□

$F$ ,  $G$  et  $\mathbf{K}$  comme définies dans la Table 3 vérifient les hypothèses du théorème 3. En effet,  $(\mathbf{L}, \mathbf{M})$  est une matrice carrée inversible de déterminant égal a 1 quelque soit la taille des données. Par conséquent, (H.2) est vérifiée.

Afin d'implémenter l'algorithme de Chambolle-Pock et résoudre le problème 21, nous calculons les opérateurs proximaux associés à  $F$  et  $G$  comme définies dans la Table 3 :

Soit  $r > 0$ . L'expression de l'opérateur proximal associé à  $F$  est déjà donnée dans l'équation 16 tandis que  $\text{prox}_{rG}$  vérifie :

$$(\text{prox}_{rG}(x))_t = \begin{cases} x_t & \text{si } t \in \{1, 2\} \\ \left( \max \left( 0, 1 - \frac{r}{|x_t|} \right) x_t \right) & \text{si } 3 \leq t \leq T \end{cases}$$

Nous présentons dans l'algorithme 3 le schéma algorithmique de Chambolle-Pock pour la résolution du problème (21).

---

**Algorithm 3** Schéma algorithmique pour l'estimation régularisée de  $\mathbf{R}$  avec pénalisation sur le vecteur des pentes

---

**Entrées:**  $\mathbf{Z}, \Phi^{\mathbf{Z}}, \lambda$

```

1: Parametres :  $\theta, \tau, \sigma, \epsilon, k_{\max}$ 
2:  $k \leftarrow 0$ 
3: on initialise  $(\mathbf{u}, \mathbf{v})^{[0]}$  en mettant toutes les pentes à 0 et l'offset à 1.
4:  $\mathbf{y}^{[0]} \leftarrow \mathbf{L}\mathbf{u}^{[0]} + \mathbf{M}\mathbf{v}^{[0]}$ 
5:  $\mathcal{A}(0) \leftarrow 1 + \epsilon$  {On initialise le critère d'itération à une valeur plus grande que  $\epsilon$ }
6: while  $\mathcal{A}(k) > \epsilon$  et  $k < k_{\max}$  do
7:    $\mathbf{R}^{[k]} \leftarrow \mathbf{L}\mathbf{u}^{[k]} + \mathbf{M}\mathbf{v}^{[k]}$ 
8:    $\mathcal{O}(k) \leftarrow D_{KL}(\mathbf{Z} | \Phi^{\mathbf{Z}} \odot \mathbf{R}^{[k]}) + \lambda \|\mathbf{v}^{[k]}\|_1$ 
9:    $\mathbf{y}^{[k+1]} \leftarrow \mathbf{y}^{[k]} + \sigma \mathbf{R}^{[k]} - \sigma \text{prox}_{\frac{1}{\sigma}F}(\frac{1}{\sigma} \mathbf{y}^{[k]} + \mathbf{R}^{[k]})$ 
10:   $(\mathbf{u}, \mathbf{v})^{[k+1]} \leftarrow \text{prox}_{\tau G}((\mathbf{u}, \mathbf{v})^{[k]} - \tau(\mathbf{L}, \mathbf{M})^\top (2\mathbf{y}^{[k+1]} - \mathbf{y}^{[k]}))$ 
11:   $\mathcal{A}(k+1) \leftarrow \max_{\max(0, k-500) < k' \leq k} \max_{1 \leq t \leq T} \frac{|R_t^{[k'+1]} - R_t^{[k'+1]}|}{R_t^{[k'+1]}}$ 
12:   $k \leftarrow k + 1$ 
13: end while
14:  $k_f \leftarrow k$ 
Sorties:  $k_f, (\hat{\mathbf{u}}, \hat{\mathbf{v}}) \leftarrow (\mathbf{u}, \mathbf{v})^{[k_f]}, \hat{\mathbf{R}} \leftarrow \mathbf{R}^{[k_f]}, \mathcal{A}, \mathcal{O}$ 

```

---

### 4.1.3 Résultats de simulations

**Données :** Pour les simulations nous utilisons les données présentées dans la figure 2 que nous normaliserons pour accélérer la convergence. En pratique nous travaillerons alors avec  $\frac{\mathbf{Z}}{\text{std}(\mathbf{Z})}$  et  $\frac{\Phi^{\mathbf{Z}}}{\text{std}(\mathbf{Z})}$ .

#### Choix des paramètres :

Pour assurer la convergence de l'algorithme de Chambolle-Pock, nous choisissons  $\tau = \sigma = \frac{0.99}{\|(\mathbf{L}, \mathbf{M})\|_2}$  et  $\theta = 1$ .

Nous fixons ensuite  $\epsilon = 10^{-8}$  et  $k_{\max} = 150000$  et nous explorons quelques niveaux de régularisation en choisissant  $\lambda \times \text{std}(\mathbf{Z})$  dans  $\{100, 300, 10000, 30000, 40000\}$ .

### Vérification de la convergence

Pour vérifier la convergence, nous visualisons pour chaque valeur de  $\lambda$  explorée l'évolution des accroissements ( $\mathcal{A}$ ) en fonction du nombre d'itérations ( $k$ ) (voir figure 9). Nous traçons aussi les itérés de la fonction objectif ( $\mathcal{O}(k)$ ) dans la figure 10.

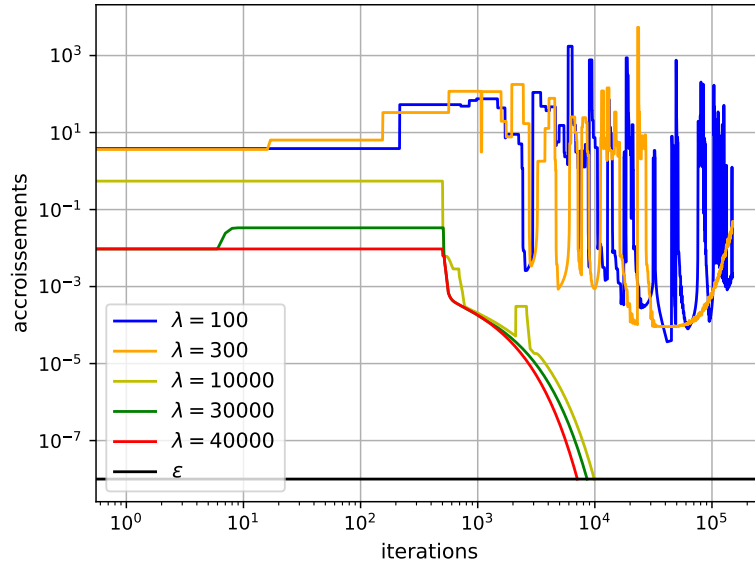


FIGURE 9 – Évolution des accroissements pour quelques valeurs de  $\lambda$

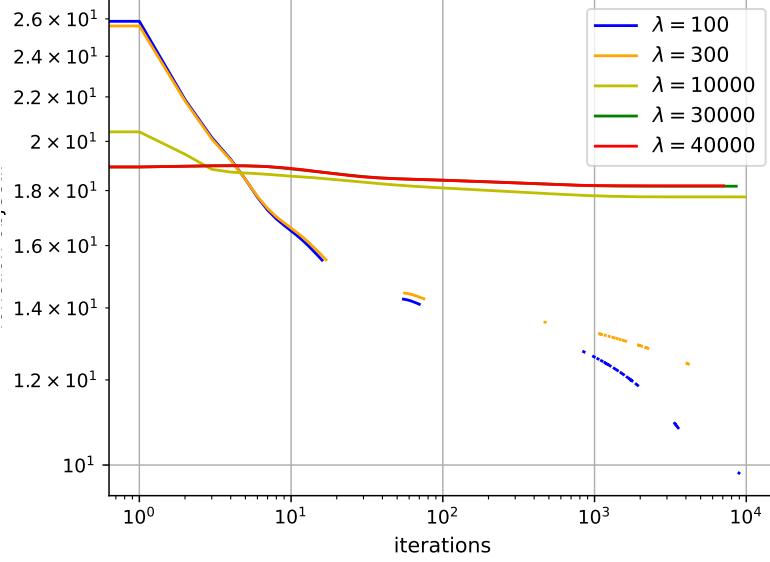


FIGURE 10 – Évolution de la fonction objectif pour quelques valeurs de  $\lambda$

**Commentaire :**

Pour  $\lambda \in \{100, 300\}$ , nous remarquons une instabilité des accroissements représentée par des fluctuations très importantes. De plus, au bout de  $k_{\max}$  itérations,  $\mathcal{A}(k_f)$  est supérieur en ordre de grandeur à  $\epsilon$ . En comparant les différentes vitesses de convergence pour les valeurs de  $\lambda$  explorées, il est clair que l'algorithme converge lentement pour  $\lambda \in \{100, 300\}$ . Nous remarquons aussi que la fonction objectif évaluée en  $(\mathbf{u}, \mathbf{v})^{[k_f]}$  vaut  $+\infty$ . La Table 4 récapitule quelques indicateurs de convergence pour les niveaux de régularisation explorés.

**Explication de la lenteur de la convergence :**

Pour  $\lambda \in \{100, 300\}$ , nous avons toujours des week-ends qui ne sont pas corrigés (voir figure 12) et par conséquent le minimiseur théorique de la fonction objectif ( $\mathbf{R}^\dagger$ ) a à priori des composantes qui sont très proches de 0 et donc de la frontière du domaine admissible de  $D_{KL}$ . L'application du seuillage doux pour le calcul des itérés de  $\mathbf{v}$  crée un biais de régularisation (voir figure 11). Il s'en suit que les composantes des itérés de  $\mathbf{R}$  qui sont censés converger vers des valeurs assez proches 0 et à l'intérieur du domaine admissible de  $D_{KL}$  sortent de ce dernier.



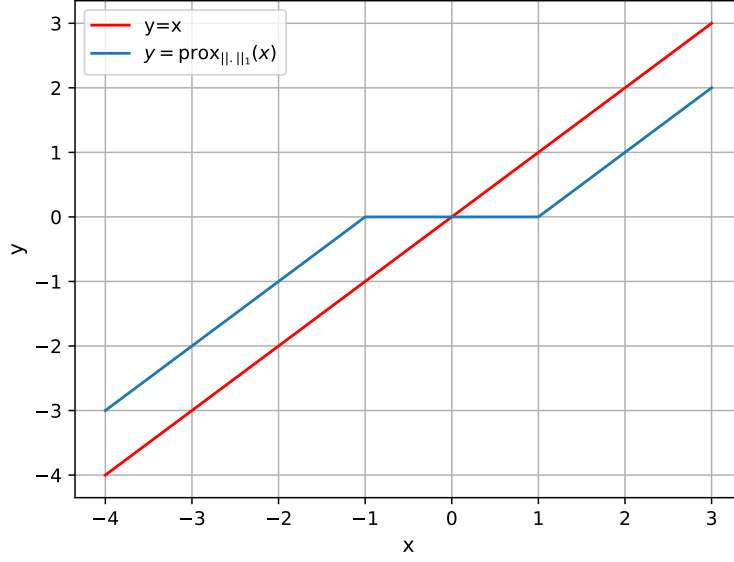


FIGURE 11 – Seuillage doux

$\lambda$	$\mathcal{O}(k_f)$	$\mathcal{A}(k_f)$	$k_f$	$\mathbf{R}^{[k_f]} \in \text{dom}(F(K.) + G)$
100	$+\infty$	0.0017	150000	Non
300	$+\infty$	0.049	150000	Non
10000	17.75829	$9.996 \times 10^{-9}$	9895	Oui
30000	18.16939	$9.996 \times 10^{-9}$	8589	Oui
40000	18.18420	$9.98 \times 10^{-9}$	7089	Oui

TABLE 4 – Tableau récapitulatif de quelques indicateurs de convergence

### Estimées de $\mathbf{R}$

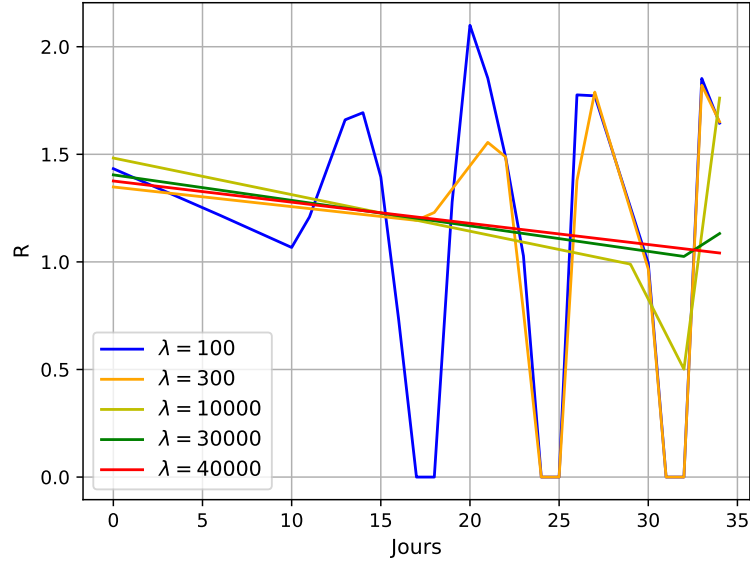


FIGURE 12 – Estimée de  $R$  pour quelques valeurs de  $\lambda$

**Commentaire :**

L'observation des estimées de  $\mathbf{R}$  met en lumière une évolution de la parcimonie sur le vecteur des pentes de gauche à droite qui s'explique par l'amplitude des données. En comparant les estimées de  $\mathbf{R}$  pour  $\lambda = 40000$  (problème à la synthèse) avec  $\lambda = 150000$  (état de l'art) nous remarquons que l'approche à la synthèse fournit une meilleure parcimonie en moins d'itérations (voir Table 5 et figure 13). Par conséquent, si nous voulons visualiser une estimée de  $\mathbf{R}$  n'ayant pas de ruptures de pentes brusques la résolution du problème (21) est plus adaptée.

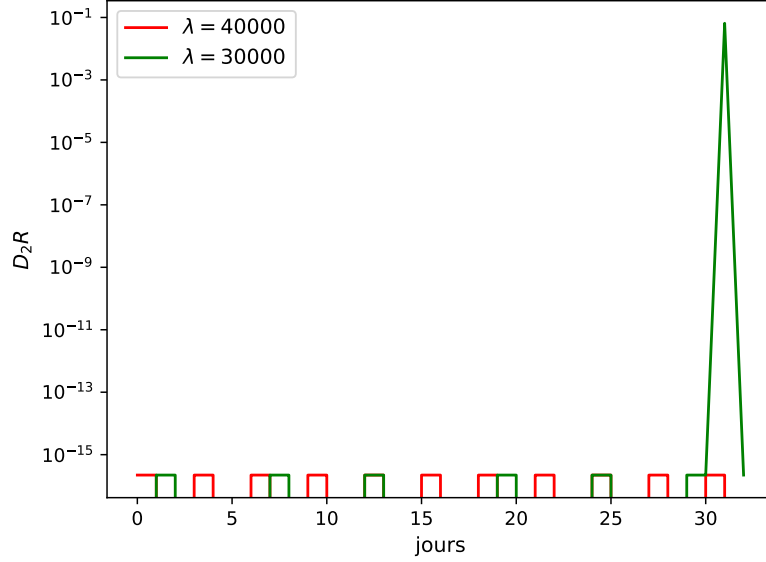


FIGURE 13 – Amplitude des estimées du Laplacien du coefficient de reproduction

Problème	$\lambda$	$k_f$	$\ \mathbf{D}_2\mathbf{R}^{[k_f]}\ _\infty$
état de l'art	150000	18335	$3.13 \times 10^{-7}$
problème à la synthèse	40000	7089	$2.22 \times 10^{-16}$

TABLE 5 – Comparaison des performances entre l'approche à la synthèse et l'état de l'art

#### 4.1.4 Limites du modèle d'optimisation

L'approche régularisée avec une pénalisation  $l_1$  crée un biais de régularisation. De plus, la parcimonie obtenue est certes meilleure par rapport à l'état de l'art mais elle n'est toujours pas exacte. Cette difficulté motive le travail de la section suivante.

## 4.2 Passage à la pénalisation $l_0$

### 4.2.1 Problème à résoudre

Pour atténuer le biais de régularisation introduit par la pénalisation  $l_1$  et obtenir de la parcimonie exacte sur le vecteur des pentes, nous allons estimer  $(\mathbf{u}, \mathbf{v})$  en considérant un terme de pénalisation plus sévère à savoir la norme  $l_0$ .

Nous cherchons alors à résoudre :

$$\min_{(\mathbf{u}, \mathbf{v}) \in \mathbb{R}^2 \times \mathbb{R}^{T-2}} D_{KL}(\mathbf{Z} | \Phi^{\mathbf{Z}} \odot (\mathbf{L}\mathbf{u} + \mathbf{M}\mathbf{v})) + \lambda \|\mathbf{v}\|_0 \quad (\text{s.c. } \|\mathbf{v}\|_\infty \leq B) \quad (24)$$

**Remarque 1.** *La résolution du problème d'optimisation (24) est plus facile que la résolution du problème d'optimisation avec pénalisation  $l_0$  sur  $\mathbf{D}_2$ . En effet, avec cette nouvelle formulation nous avons pu contourner la présence de l'opérateur Laplacien au niveau du terme de pénalisation.*

### 4.2.2 Méthode de résolution

La méthode de résolution que nous décrivons ci-dessous est inspirée de la thèse [7].

L'utilisation des outils de l'optimisation convexe non lisse n'est pas adaptée pour résoudre (24) vu que  $\|\cdot\|_0$  n'est pas convexe. Une première approche naïve et assez systématique est de partitionner  $\mathbb{R}^{T-2}$  en imposant à chaque fois des conditions de nullité sur les coordonnées de  $\mathbf{v}$ , minimiser la restriction de la fonction objectif à chaque partition et choisir la meilleure solution. Cependant, cette méthode est très coûteuse en temps de calcul puisqu'il faudra résoudre  $2^{T-2}$  sous-problèmes. Pour contourner cette difficulté nous allons utiliser la méthode algorithmique Branch-and-Bound (séparation et évaluation en français).

#### Principe de résolution général :

L'algorithme se base principalement sur le parcours judicieux d'un ensemble de solutions réalisables en alternant deux étapes :

- la séparation qui consiste à diviser selon une règle de décision un problème complexe que nous assimilons à un noeud parent en sous problèmes que nous assimilons à des noeuds fils tout en constituant un arbre de recherche,
- l'évaluation qui consiste à déterminer les caractéristiques d'un noeud pour conclure quant à son élagage et par conséquent réduire l'espace de recherche de la solution optimale.

#### Principe de résolution adapté à notre problème d'optimisation :

### Première partie : La division en sous problèmes

La division d'un problème en sous problèmes se fait en imposant des contraintes de nullité sur les coordonnées du vecteurs de pentes  $\mathbf{v}$ . Pour illustrer ce propos, nous définissons ci-dessous une partition de  $\llbracket 1, T-2 \rrbracket$  qui nous permettra de déterminer en ligne et pour chaque sous problème les conditions de nullité imposées au cours du processus d'exploration de l'arbre.

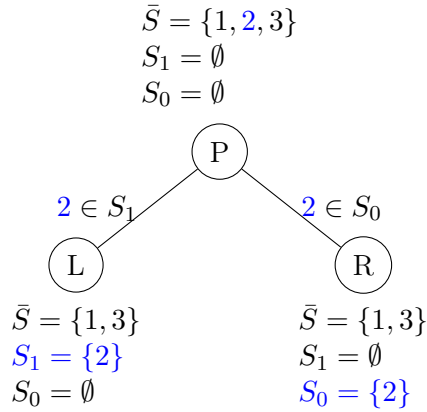
Considérons  $\bar{S}, S_0, S_1, T \geq 3$  tq :

$$S_1 = \{t \in \llbracket 1, T-2 \rrbracket \text{ tq } v_t \in \mathbb{R}^*\}$$

$$S_0 = \{t \in \llbracket 1, T-2 \rrbracket \text{ tq } v_t = 0\}$$

$$\bar{S} = (S_1 \cup S_0) \cap \llbracket 1, T-2 \rrbracket$$

Nous illustrons dans le graphe ci-dessous le principe de division et la construction de la première couche de l'arbre à explorer avec un jeu de données de taille  $T = 5$ .



**Remarque 2.** Dans l'exemple ci-dessus, nous avons choisi aléatoirement 2 comme indice à contraindre. Nous définirons plus loin une règle de décision pour choisir l'indice  $i \in \llbracket 1, T-2 \rrbracket$  à mettre dans  $S_1$  ou  $S_0$ .

### Deuxième partie : Les caractéristiques d'un sous problème

Chaque sous problème est caractérisé par cinq quantités : L'état des contraintes  $\bar{S}$ ,  $S_1$  et  $S_0$  comme illustré dans le graphe, une borne inférieure et une borne supérieure.

La borne inférieure que nous notons LB et qui consiste à considérer un mino- rant pour la solution d'un sous-problème. Pour l'obtenir, nous considérons

une relaxation continue et convexe de la norme  $l_0$  afin d'obtenir un problème convexe que nous résoudrons facilement. Nous nous contenterons à cette étape du rapport de mentionner le résultat (25) qui servira de base pour construire LB. Nous détaillerons par la suite dans une section dédiée le calcul de cette borne.

$$\|\mathbf{v}\|_\infty \leq B \Rightarrow \frac{1}{B} \|\mathbf{v}\|_1 \leq \|\mathbf{v}\|_0 \quad (25)$$

La borne supérieure que nous notons UB est obtenue en résolvant le sous-problème sous-condition de mettre tous les coefficients indexés par  $\bar{S}$  à 0. Elle représente alors une borne supérieure sur la valeur optimale fournie par la résolution du sous-problème. De plus, la borne supérieure la plus optimale (plus petite) obtenue au cours de l'exécution du Branch-and Bound représente la solution optimale du problème général. Nous détaillerons par la suite dans une section dédiée le calcul de UB.

Pour élaguer un noeud nous vérifions si la borne inférieure qui lui est associée est supérieure à la borne supérieure la plus optimale obtenue à cette étape de l'exploration. Le schéma algorithmique 4 décrit le fonctionnement du Branch-and-Bound.

---

**Algorithm 4** Schéma algorithmique du Branch-and-Bound
 

---

**Entrées:**  $\mathbf{Z}, \Phi^{\mathbf{Z}}, \lambda, B$

```

1: On considère le noeud initial ( $N^{initial}$ ) caractérisé par  $S_0 = S_1 = \emptyset$ 
2: On calcule  $LB^{N^{initial}}$  et  $UB^{N^{initial}}$  {borne inf et borne sup pour le noeud
   initial }
3:  $UB^{ref} \leftarrow UB^{N^{initial}}$  {On stocke la borne sup dans la variable à mettre
   a jour}
4: Liste  $\leftarrow [N^{initial}]$  {On stocke le noeud initial dans Liste pour l'explorer}
5: while  $L \neq \emptyset$  do
6:   On prend le dernier élément de  $L$  qu'on note  $N$  pour l'explorer.
7:   On choisit un indice pour construire les deux fils de  $N$  {L'algorithme
   7 présente la démarche pour effectuer ce choix }.
8:   On calcule  $LB^{N^{filsgauche}}$  et  $UB^{N^{filsgauche}}$ 
9:   if  $LB^{N^{filsgauche}} < UB^{ref}$  then
10:    On ajoute  $N^{filsgauche}$  à Liste
11:    if  $UB^{N^{filsgauche}} < UB^{ref}$  then
12:       $UB^{ref} \leftarrow UB^{N^{filsgauche}}$ 
13:    end if
14:  end if
15:  On calcule  $LB^{N^{filsdroit}}$  et  $UB^{N^{filsdroit}}$ 
16:  if  $LB^{N^{filsdroit}} < UB^{ref}$  then
17:    On ajoute  $N^{filsdroit}$  à Liste
18:    if  $UB^{N^{filsdroit}} < UB^{ref}$  then
19:       $UB^{ref} \leftarrow UB^{N^{filsdroit}}$ 
20:    end if
21:  end if
22: end while
Sorties:  $UB^{ref}$ 

```

---

**Définition d'un sous problème :**

Soient  $C = \{n_1, \dots, n_{|C|}\}$  et  $D = \{m_1, \dots, m_{|D|}\}$  deux sous ensembles ordonnés de  $\mathbb{N}$  tq  $|C| < \infty$  et  $|D| < \infty$ .

Soit  $v$  un vecteur de taille  $T \geq |C| + |D|$ .

Nous définissons  $y = v_{C \cup D}$  comme étant le vecteur de taille  $|C| + |D|$  tq :

$$\begin{aligned} \forall i \in [1, |C|] : y_i &= v_{n_i} \\ \forall i \in [|C| + 1, |C| + |D|] : y_i &= v_{m_{i-|C|}} \end{aligned}$$

Soient  $S, S_0, S_1$  l'état des contraintes a une étape donnée de la construction de l'arbre. Nous définissons un sous problème par :

$$(P) : \min_{(\mathbf{u}, \mathbf{v}_{S_1 \cup \bar{S}}) \in \mathbb{R}^2 \times \mathbb{R}^{|S_1| + |\bar{S}|}} D_{KL}(\mathbf{Z} | \Phi^{\mathbf{Z}} \odot (\mathbf{L}\mathbf{u} + \mathbf{M}_{S_1 \cup \bar{S}} \mathbf{v}_{S_1 \cup \bar{S}})) + \lambda \|\mathbf{v}_{\bar{S}}\|_0 + \lambda |S_1| \quad (\text{s.c. } \|\mathbf{v}_{S_1 \cup \bar{S}}\|_{\infty} \leq B)$$

(26)

**Calcul de la borne inférieure (LB) associé a (P) :**

Pour calculer la borne inférieure nous exploitons l'équation (25). En effet :

$$\begin{aligned} \|\mathbf{v}_{S_1 \cup \bar{S}}\|_{\infty} \leq B &\Rightarrow \|\mathbf{v}_{\bar{S}}\|_{\infty} \leq B \\ &\Rightarrow \frac{1}{B} \|\mathbf{v}_{\bar{S}}\|_1 \leq \|\mathbf{v}_{\bar{S}}\|_0 \end{aligned}$$

Nous cherchons alors a résoudre :

$$(P^{LB}) : \begin{cases} \min_{(\mathbf{u}, \mathbf{v}_{S_1 \cup \bar{S}}) \in \mathbb{R}^2 \times \mathbb{R}^{|S_1| + |\bar{S}|}} D_{KL}(\mathbf{Z} | \Phi^{\mathbf{Z}} \odot (\mathbf{L}\mathbf{u} + \mathbf{M}_{S_1 \cup \bar{S}} \mathbf{v}_{S_1 \cup \bar{S}})) + \mu \|\mathbf{v}_{\bar{S}}\|_1 + \lambda |S_1| \\ \text{s.c. } \|\mathbf{v}_{S_1 \cup \bar{S}}\|_{\infty} \leq B \end{cases}$$

(27)

avec  $\mu = \frac{\lambda}{B}$

Le problème (27) est un problème sous-contrainte. Nous le réécrivons sous la forme d'un problème d'optimisation de type  $\min_{x \in \mathcal{X}} F(\mathbf{K}x) + G(x)$  en exprimant la contrainte de bord sous forme d'une indicatrice.

$$(P^{LB}) : \min_{(\mathbf{u}, \mathbf{v}_{S_1 \cup \bar{S}}) \in \mathbb{R}^2 \times \mathbb{R}^{|S_1| + |\bar{S}|}} D_{KL}(\mathbf{Z} | \Phi^{\mathbf{Z}} \odot (\mathbf{L}\mathbf{u} + \mathbf{M}_{S_1 \cup \bar{S}} \mathbf{v}_{S_1 \cup \bar{S}})) + \mu \|\mathbf{v}_{\bar{S}}\|_1 + \iota_{\mathcal{B}}(\mathbf{v}_{S_1 \cup \bar{S}}) + \lambda |S_1|$$

(28)

avec  $\mathcal{B} = [-B, B]^{|S_1| + |\bar{S}|}$  et :

$$\iota_{\mathcal{B}}(x) = \begin{cases} 0 & \text{si } x \in \mathcal{B} \\ \infty & \text{sinon} \end{cases}$$

Le problème de minimisation (22) s'identifie au problème général d'optimisation convexe non lisse (9). La Table 6 fait le lien entre les deux problèmes.



Problème général	$(P^{LB})$
$\mathcal{X}$	$\mathbb{R}^{2+ S_1 + \bar{S} }$
$\mathcal{Y}$	$\mathbb{R}^T$
$x$	$(\mathbf{u}, \mathbf{v}_{S_1 \cup \bar{S}})$
$F$	$D_{KL}(\mathbf{Z}   \Phi^{\mathbf{Z}} \odot .)$
$G$	$(\mathbf{u}, \mathbf{v}_{S_1 \cup \bar{S}}) \rightarrow \frac{\lambda}{B} \ \mathbf{v}_{\bar{S}}\ _1 + \iota_{\mathcal{B}}(\mathbf{v}_{S_1 \cup \bar{S}}) + \lambda  S_1 $
$\mathbf{K}$	$(\mathbf{L}, \mathbf{M}_{S_1 \cup \bar{S}})$

TABLE 6 – Identification des composantes du problème du calcul de la borne inférieure à celle du problème 9

Remarquons que  $F$  et  $G$  ainsi définies vérifient (H.1). De plus,  $(\mathbf{L}, \mathbf{M}_{S_1 \cup \bar{S}})$  est de rang plein car  $(\mathbf{L}, \mathbf{M})$  est inversible. Le théorème 3 s'applique et (H.2) est vérifiée.

Afin d'implémenter l'algorithme de Chambolle-Pock pour le calcul de LB nous donnons les expressions des opérateurs proximaux associés à  $F$  et  $G$  comme définies dans (Table 4).

L'expression de  $\text{prox}_{rF}$  est fournie dans l'équation 16. Pour calculer  $\text{prox}_{rG}$ , nous exploiterons la séparabilité de la fonction  $G$ . En effet :

$$\begin{aligned}
G(\mathbf{u}, \mathbf{v}_{S_1 \cup \bar{S}}) &= \frac{\lambda}{B} \|\mathbf{v}_{\bar{S}}\|_1 + \iota_{\mathcal{B}}(\mathbf{v}_{S_1 \cup \bar{S}}) + \lambda |S_1| \\
&= \frac{\lambda}{B} \|\mathbf{v}_{\bar{S}}\| + \iota_{[-B, B]^{|\bar{S}|}}(\mathbf{v}_{\bar{S}}) + \iota_{[-B, B]^{|\bar{S}|}}(\mathbf{v}_{S_1}) \\
&= (\iota_{[-B, B]^{|\bar{S}|}} + \frac{\lambda}{B} \|\cdot\|_1)(\mathbf{v}_{\bar{S}}) + \iota_{[-B, B]^{|\bar{S}|}}(\mathbf{v}_{S_1})
\end{aligned}$$

Grâce à la séparabilité de  $\|\cdot\|_2$  nous pouvons calculer  $\text{prox}_{rG}$  en calculant  $\text{prox}_{r\iota_{\mathcal{P}}}$  et  $\text{prox}_{r(\iota_{\mathcal{P}} + \frac{\lambda}{B} \|\cdot\|_1)}$ .

**Proposition 5.** Soit  $r > 0$ ,  $\lambda > 0$ ,  $n > 1$ ,  $y \in \mathbb{R}^n$  et  $\mathcal{P} = [-B, B]^n$  la boule de rayon  $B$  associées à  $\|\cdot\|_{\infty}$ . On pose  $\beta = \frac{r\lambda}{B}$  alors :

$$\forall t \in \llbracket 1, n \rrbracket : (\text{prox}_{r(\iota_{\mathcal{P}})}(y))_t = \begin{cases} -B & \text{si } y_t < -B \\ B & \text{si } y_t > B \\ y_t & \text{si } -B \leq y_t \leq B \end{cases} \quad (29)$$

$$\forall t \in \llbracket 1, n \rrbracket : (\text{prox}_{r(\iota_{\mathcal{P}} + \frac{\lambda}{B} \|\cdot\|_1)}(y))_t = \begin{cases} B & \text{si } y_t \geq \beta + B \\ y_t - \beta & \text{si } \beta \leq y_t \leq \beta + B \\ -B & \text{si } y_t \leq -\beta - B \\ y_t + \beta & \text{si } -\beta - B \leq y_t \leq -\beta \\ 0 & \text{si } |y_t| \leq \beta \end{cases} \quad (30)$$

Par conséquent,

$$(\text{prox}_{rG}(\mathbf{u}, \mathbf{v}_{S_1 \cup \bar{S}}))_t = \begin{cases} u_t & \text{si } t = 1, 2 \\ (\text{prox}_{r\iota_{[-B, B] | S_1|}}(v_{S_1}))_t & \text{si } S_1 \neq \emptyset \text{ et } t = 3, \dots, 2 + |S_1| \\ (\text{prox}_{r(\iota_{[-B, B] | S|} + \frac{\lambda}{B} \|\cdot\|_1)}(v_S))_t & \text{si } S \neq \emptyset \text{ et } t = 3 + |S_1|, \dots, 2 + |S| + |S_1| \end{cases}$$

---

**Algorithm 5** Schéma algorithmique pour le calcul de la borne inférieure

---

**Entrées:**  $\mathbf{Z}, \Phi^{\mathbf{Z}}, \lambda, S_1, S_0, \bar{S}$

1: **Parametres** :  $\theta, \tau, \sigma, \epsilon, k_{max}$   
2:  $k \leftarrow 0$   
3: on initialise  $(\mathbf{u}, \mathbf{v}_{S_1 \cup \bar{S}})^{[0]}$  en mettant toutes les pentes à 0 et l'offset a 1.  
4:  $\mathbf{y}^{[0]} \leftarrow (\mathbf{L}, \mathbf{M}_{S_1 \cup \bar{S}})(\mathbf{u}, \mathbf{v}_{S_1 \cup \bar{S}})^{[0]}$   
5:  $\mathcal{A}(0) \leftarrow 1 + \epsilon$   
6: **while**  $\mathcal{A}(k) > \epsilon$  et  $k < k_{max}$  **do**  
7:    $\mathbf{R}^{[k]} = (\mathbf{L}, \mathbf{M}_{S_1 \cup \bar{S}})(\mathbf{u}, \mathbf{v}_{S_1 \cup \bar{S}})^{[k]}$   
8:    $\mathbf{y}^{[k+1]} \leftarrow \mathbf{y}^{[k]} + \sigma \mathbf{R}^{[k]} - \sigma \text{prox}_{\frac{1}{\sigma} F}(\frac{1}{\sigma} \mathbf{y}^{[k]} + \mathbf{R}^{[k]})$   
9:    $(\mathbf{u}, \mathbf{v}_{S_1 \cup \bar{S}})^{[k+1]} \leftarrow \text{prox}_{\tau G}((\mathbf{u}, \mathbf{v}_{S_1 \cup \bar{S}})^{[k]} - \tau(\mathbf{L}, \mathbf{M}_{S_1 \cup \bar{S}})^\top (2\mathbf{y}^{[k+1]} - \mathbf{y}^{[k]}))$   
10:    $\mathcal{A}(k+1) \leftarrow \max_{\max(0, k-500) < k' \leq k} \max_{1 \leq t \leq T} \frac{|R_t^{[k'+1]} - R_t^{[k']}|}{R_t^{[k'']}}$   
11:    $k \leftarrow k + 1$   
12: **end while**  
13:  $k_f \leftarrow k$   
14:  $LB \leftarrow D_{KL}(\mathbf{Z} | \Phi^{\mathbf{Z}} \odot (\mathbf{L} \mathbf{u}^{[k_f]} + \mathbf{M}_{S_1 \cup \bar{S}} \mathbf{v}_{S_1 \cup \bar{S}}^{[k_f]})) + \mu \|\mathbf{v}_{\bar{S}}^{[k_f]}\|_1 + \iota_{\mathcal{B}}(\mathbf{v}_{S_1 \cup \bar{S}}^{[k_f]}) + \lambda |S_1|$   
**Sorties:**  $k_f, (\hat{\mathbf{u}}, \hat{\mathbf{v}}_{S_1 \cup \bar{S}}) \leftarrow (\mathbf{u}, \mathbf{v}_{S_1 \cup \bar{S}})^{[k_f]}, \hat{\mathbf{R}} \leftarrow \mathbf{R}^{[k_f]}, \mathcal{A},$

---

Le calcul de la borne inférieure en utilisant cet algorithme met en lumière quelques difficultés à l'image de celle rencontrées pendant la résolution du problème de parcimonie sur le vecteur des pentes en utilisant la pénalisation  $l_1$ . En effet, en ayant  $\frac{\lambda}{B}$  proche de 0 la convergence est lente (ordre de grandeur de  $10^6$  itérations). Ainsi, en fixant un nombre d'itérations maximal raisonnable  $k_{max} = 150000$  pour que le Branch-and-Bound puisse converger, nous obtenons en fin d'exécution  $\mathbf{R}^{[k_f]}$  à l'extérieur du domaine admissible de la fonction objectif à minimiser. Il s'en suit que nous obtenons  $LB = +\infty$  (voir ligne 14 algorithme 5). Par conséquent, nous risquons d'élaguer à tort des noeuds. Pour remédier à ce problème et mettre en place un compromis rapidité-précision nous introduisons une relaxation de la divergence de Kullback-Leibler ( $D_{KL}$ ) que nous notons  $D_{KL}^{etendue}$  et qui vérifie :

$$D_{KL}^{etendue} = \sum_{t=1}^T d_{KL}^{etendue}(Z_t | \Phi_t^Z * R_t)$$

avec :

$$d_{KL}^{etendue}(Z_t | \Phi_t^Z * R_t) = \begin{cases} Z_t \log(\frac{Z_t}{R_t \Phi_t^Z}) + R_t \Phi_t^Z - Z_t & \text{si } Z_t > 0 \text{ et } R_t > 0 \\ |R_t \Phi_t^Z| & \text{sinon} \end{cases} \quad (31)$$

**Remarque 3.**  $D_{KL}^{etendue}$  coïncide avec  $D_{KL}$  sur son domaine admissible et consiste une bonne approximation à l'extérieur de ce dernier. L'idée est de construire une fonction objectif qui évolue de façon lisse si  $\mathbf{Z} = 0$ .

Nous calculerons en pratique  $LB$  à partir de l'expression suivante :

$$LB = D_{KL}^{etendue}(\mathbf{Z} | \Phi^{\mathbf{Z}} \odot (\mathbf{L}\mathbf{u}^{[k_f]} + \mathbf{M}_{S_1 \cup \bar{S}} \mathbf{v}_{S_1 \cup \bar{S}}^{[k_f]})) + \mu \|\mathbf{v}_S^{[k_f]}\|_1 + \iota_{\mathcal{B}}(\mathbf{v}_{S_1 \cup \bar{S}}^{[k_f]}) + \lambda |S_1|$$

#### Calcul de la borne supérieure (UB)

Pour calculer la borne supérieure on met à 0 toutes les coordonnées indexées par  $\bar{S}$  et on optimise. Nous cherchons donc à résoudre :

$$(P^{UB}) : \min_{(\mathbf{u}, \mathbf{v}_{S_1}) \in \mathbb{R}^2 \times \mathbb{R}^{|S_1|}} D_{KL}(\mathbf{Z} | \Phi^{\mathbf{Z}} \odot (\mathbf{L}\mathbf{u} + \mathbf{M}_{S_1} \mathbf{v}_{S_1})) + \iota_{\mathcal{B}}(\mathbf{v}_{S_1}) + \lambda |S_1| \quad (32)$$

Le problème 24 s'identifie au problème général d'optimisation convexe non lisse. La comparaison est détaillée dans la Table 3.

Problème général	$(P^{UB})$
$\mathcal{X}$	$\mathbb{R}^{2+ S_1 }$
$\mathcal{Y}$	$\mathbb{R}^T$
$x$	$(\mathbf{u}, \mathbf{v}_{S_1})$
$F$	$D_{KL}(\mathbf{Z}   \Phi^{\mathbf{Z}} \odot .)$
$G$	$(\mathbf{u}, \mathbf{v}_{S_1}) \rightarrow +\iota_{\mathcal{B}}(\mathbf{v}_{S_1}) + \lambda  S_1 $
$\mathbf{K}$	$(\mathbf{L}, \mathbf{M}_{S_1})$

TABLE 7 – Identification des composantes du problème de calcul de la borne supérieure à celle du problème 9

Remarquons que  $F$  et  $G$  ainsi définies vérifient (H.1). De plus,  $(\mathbf{L}, \mathbf{M}_{S_1})$  est de rang plein car  $(\mathbf{L}, \mathbf{M})$  inversible. Par suite, le théorème 2 s'applique et (H.2) est vérifiée.

Calculons les opérateurs proximaux associés a  $F$  et  $G$  comme définies dans la Table 5 :

Soit  $r > 0$ .

L'expression de  $prox_{rF}$  est fournie dans l'équation (14) et  $prox_{rG}$  vérifie :

$$(prox_{rG}(\mathbf{u}, \mathbf{v}_{S_1}))_t = \begin{cases} u_t & \text{si } t = 1, 2 \\ (prox_{\tau \iota_{[-B, B]^{|S_1|}}}(\mathbf{v}_{S_1}))_t & \text{si } |S_1| \neq \emptyset \text{ et } t = 3, \dots, |S_1| + 2 \end{cases}$$

---

**Algorithm 6** Schéma algorithmique pour le calcul de la borne supérieure

---

**Entrées:**  $\mathbf{Z}, \Phi^{\mathbf{Z}}, \lambda, S_1, S_0, \bar{S}$

1: **Parametres** :  $\theta, \tau, \sigma, \epsilon, k_{max}$   
2:  $k \leftarrow 0$   
3: on initialise  $(\mathbf{u}, \mathbf{v}_{S_1})^{[0]}$  en mettant toutes les pentes à 0 et l'offset à 1.  
4:  $\mathbf{y}^{[0]} \leftarrow (\mathbf{L}, \mathbf{M}_{S_1})(\mathbf{u}, \mathbf{v}_{S_1})^{[0]}$   
5:  $\mathcal{A}(0) \leftarrow 1 + \epsilon$   
6: **while**  $\mathcal{A}(k) > \epsilon$  et  $k < k_{max}$  **do**  
7:    $\mathbf{R}^{[k]} = (\mathbf{L}, \mathbf{M}_{S_1})(\mathbf{u}, \mathbf{v}_{S_1})^{[k]}$   
8:    $\mathbf{y}^{[k+1]} \leftarrow \mathbf{y}^{[k]} + \sigma \mathbf{R}^{[k]} - \sigma prox_{\frac{1}{\sigma} F}(\frac{1}{\sigma} \mathbf{y}^{[k]} + \mathbf{R}^{[k]})$   
9:    $(\mathbf{u}, \mathbf{v}_{S_1})^{[k+1]} \leftarrow prox_{\tau G}((\mathbf{u}, \mathbf{v}_{S_1})^{[k]} - \tau(\mathbf{L}, \mathbf{M}_{S_1})^\top (2\mathbf{y}^{[k+1]} - \mathbf{y}^{[k]}))$   
10:    $\mathcal{A}(k+1) \leftarrow \max_{\max(0, k-500) < k' \leq k} \max_{1 \leq t \leq T} \frac{|R_t^{[k'+1]} - R_t^{[k']}|}{R_t^{[k'()]}}$   
11:    $k \leftarrow k + 1$   
12: **end while**  
13:  $k_f \leftarrow k$   
14:  $UB \rightarrow D_{KL}^{etendue}(\mathbf{Z} | \Phi^{\mathbf{Z}} \odot (\mathbf{L}\mathbf{u}^{[k_f]} + \mathbf{M}_{S_1}\mathbf{v}_{S_1}^{[k_f]})) + \iota_{\mathcal{B}}(\mathbf{v}_{S_1}^{[k_f]}) + \lambda|S_1|$   
**Sorties:**  $k_f, (\hat{\mathbf{u}}, \hat{\mathbf{v}}_{S_1}) \leftarrow (\mathbf{u}, \mathbf{v}_{S_1})^{[k_f]}, \hat{\mathbf{R}} \leftarrow \mathbf{R}^{[k_f]}, \mathcal{A}$

---

**Choix de l'indice de la composante à contraindre [2]**

Soit  $(P)$  un noeud dans l'arbre d'exploration du Branch-and Bound.

Pour explorer  $(P)$ , nous utilisons la méthode amplitude maximale (AM) qui consiste à considérer l'indice de la plus grande composante en valeur absolue du vecteur  $\mathbf{v}_{\bar{S}}^{LB}$  obtenu en sortie de l'algorithme de calcul de la borne inférieure associé à ce noeud. La démarche algorithmique est décrite dans l'algorithme 7.

---

**Algorithm 7** Schéma algorithmique représentant la règle de décision pour construire l'arbre de recherche

---

— **Entrées:** Noeud

$indice \leftarrow argmax(|\mathbf{v}_{\bar{S}}^{LB}|)$

**Sorties:** indice

---

### 4.2.3 Résultats de simulations

#### Vérification du fonctionnement de l'algorithme Branch-and-Bound : Données

Pour accélérer la vérification, nous nous sommes contentés de la restriction de nos données aux 10 premiers jours (figure 14) que nous normaliserons. En pratique nous utilisons alors  $(\frac{\mathbf{Z}_{\llbracket 1,10 \rrbracket}}{\text{std}(\mathbf{Z}_{\llbracket 1,10 \rrbracket})}$  et  $\frac{\Phi_{\llbracket 1,10 \rrbracket}^Z}{\text{std}(\mathbf{Z}_{\llbracket 1,10 \rrbracket})}$ ).

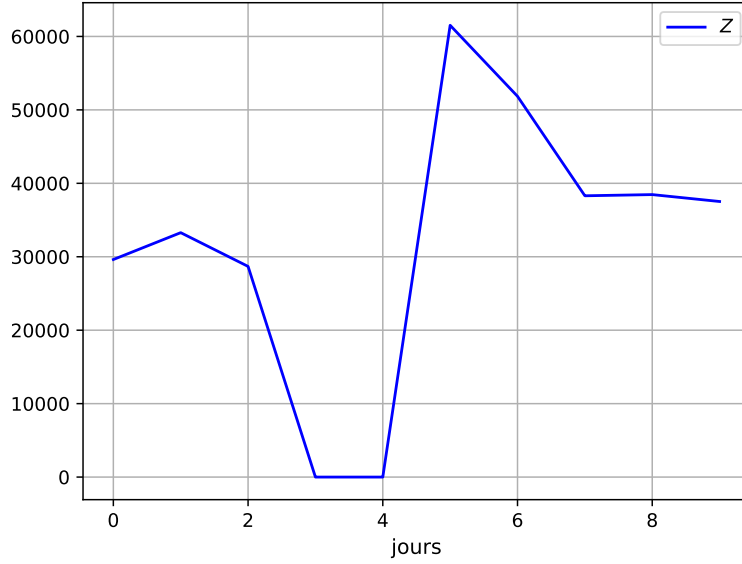


FIGURE 14 – Données restreintes aux 10 premiers jours

**Choix des parametres :** Nous choisissons  $\epsilon = 10^{-12}$  et  $k_{max} = 150000$  et  $B = 1.5 \|\mathbf{v}^{MLE}\|_{\infty} = 1.5 \times 38$ .

#### Protocole de verification

Pour s'assurer du bon fonctionnement du Branch-and-Bound, nous le comparons avec une méthode systématique qui consiste à chercher la meilleure solution faisable parmi toutes les configurations possibles de  $S_1$ ,  $S_0$  et  $\bar{S}$ . Pour cela, nous prenons  $\lambda = 1$ . Avec ce niveau de régularisation et en appliquant la méthode de recherche naïve, nous obtenons au bout de 6135 secondes (1h40min) d'exécution une estimée de  $\mathbf{R}$  ayant deux ruptures de pentes. Le Branch-and-Bound fournit les mêmes estimées et les résultats obtenus sont résumés dans la Table 8.

type de vérification	$S_1$ du noeud optimal	$S_0$ du noeud optimal	UB	temps exécution
Systématique	{4,5 }	{1,2,3,6,7,8}	2.767882	6135
Branch-and Bound	{4,5 }	{1,2,3,6,7,8}	2.767882	348

TABLE 8 – Comparaison entre Branch-and-Bound et méthode systématique pour  $\lambda = 1$

### Complexité du Branch-and-Bound pour quelques niveaux de régularisation

Nous présentons dans la Table 9 les performances de l’algorithme Branch-and-Bound pour différents niveaux de régularisation de  $\lambda$  a savoir le minimum de la fonction objectif que nous notons  $UB^{optimal}$ , le nombre de pentes non nulles que nous notons  $\|\mathbf{v}\|_0$ , le nombre de noeuds explorés, le nombre de noeud élagués et enfin le temps d’exécution en secondes ( $t_f$ ).

$\lambda$	$UB^{optimal}$	$\ \mathbf{v}\ _0$	#noeuds explorés	#noeuds élagués	$t_f$
0.1	0.44	4	13	6	1114
0.31	1.25	3	11	8	730
1	2.7678	2	11	8	348
3.16	4.415	0	5	6	102

TABLE 9 – Performances de l’algorithme Branch-and-Bound pour quelques niveaux de régularisation

### Commentaire :

L’analyse des performances du Branch-and-bound montre que le temps d’exécution est une fonction décroissante de  $\lambda$ . Si nous supposons qu’au plus 2 ruptures de pentes sont suffisantes pour conclure quant à l’évolution de la pandémie alors pour des données de taille 10 le temps d’exécution ne dépasse pas 350 secondes soit au plus 6 minutes. Nous remarquons aussi que nous obtenons de la parcimonie exacte sur  $\hat{\mathbf{v}}$ .

### Limites du modèle :

Avec des données de taille plus importante l’algorithme prend beaucoup plus de temps à converger surtout si nous désirons visualiser quelques ruptures de pentes. Nous expliquons ce propos en établissant une comparaison entre deux situations. Une première où nous utilisons nos données d’origine (présentées dans figure 2 et une deuxième avec les données de taille 10 (présentées dans figure 14).

Le calcul de la borne inférieure et de la borne supérieure sont ceux qui prennent le plus de temps lors de l’exécution du Branch-and-Bound. De plus, plus  $\mu = \frac{\lambda}{B}$  est petit plus le temps de calcul de la borne inférieure est important. Par conséquent, le choix de  $\lambda$  et  $B$  déterminera la vitesse de convergence du Branch-and-Bound. Supposons a présent que nous désirons visualiser quelques ruptures de pentes alors avec nos données de taille 35

alors nous devons choisir  $B = 1.5 \times \|\mathbf{v}^{MLE}\|_\infty = 1.5 \times 1248$ . Il est donc au moins 30 fois plus grand que  $B$  choisi pour le protocole de vérification (données de taille 10). En prenant en compte que la norme  $l_0$  est sévère en terme de régularisation il faudra choisir  $\lambda$  petit et donc  $\mu$  au moins  $10^{-3}$  fois plus petit que  $\lambda$  choisi.

### 4.3 Comparaison entre les résultats fournis dans l'état de l'art et les contributions

Nous comparons dans ce qui suit les résultats fournis par la résolution du problème (6) et des problèmes (21) et (24). Pour évaluer la pertinence des contributions apportées, nous choisissons une gamme de  $\lambda$  donnant 2 ruptures de pentes pour chaque problème. Nous nous contenterons d'utiliser les données de taille 10 présentées dans la figure 14.

La comparaison se fera par rapport aux performances des algorithmes d'estimation à savoir, la fidélité au modèle épidémiologique et le temps d'exécution. En ce qui concerne la parcimonie, nous avons montré plus haut que l'approche régularisée avec une pénalisation  $l_0$  permet d'obtenir de la parcimonie exacte sur les estimées du vecteur des pentes.

Dans la Figure 15 nous traçons  $\hat{\mathbf{R}}$  obtenu par l'estimation régularisée avec pénalisation  $l_1$  sur le Laplacien pour trois valeurs de  $\lambda$  donnant tous 2 ruptures de pentes brusques.

Nous effectuons la même démarche pour l'estimation régularisée avec pénalisation  $l_1$  (Figure 16) sur le vecteur des pentes et pour l'estimation avec pénalisation  $l_0$  (Figure 17).

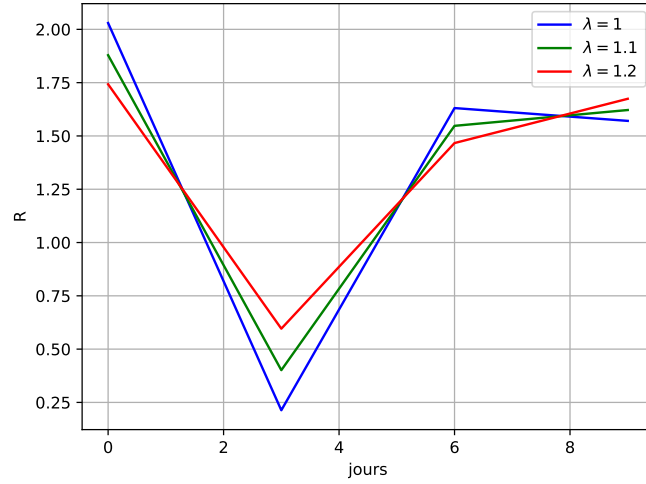


FIGURE 15 – Estimées du coefficient de reproduction à partir du problème d’optimisation régularisée décrit dans l’état de l’art pour une gamme de  $\lambda$  donnant 2 ruptures de pentes

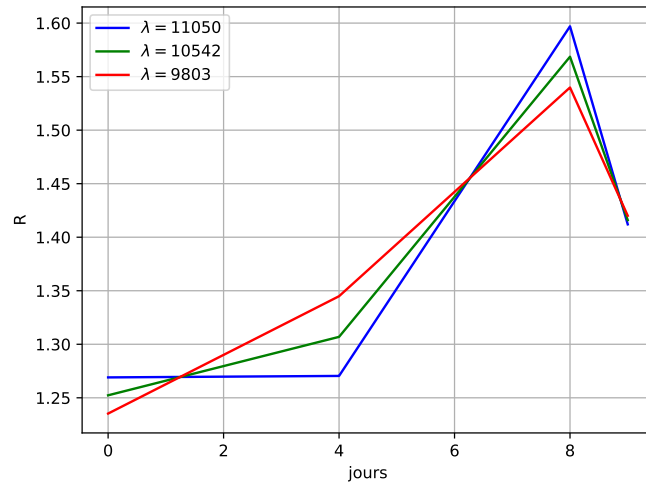


FIGURE 16 – Estimées du coefficient de reproduction obtenues par résolution du problème d’estimation régularisée avec pénalisation  $l_1$  sur le vecteur des pentes pour une gamme de  $\lambda$  donnant 2 ruptures de pentes



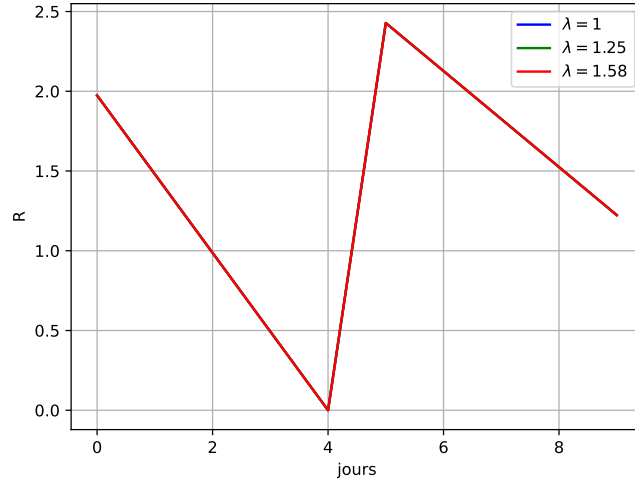


FIGURE 17 – Estimées du coefficient de reproduction obtenues par résolution du problème d’estimation régularisée avec pénalisation  $l_0$  sur le vecteur des pentes pour une gamme de  $\lambda$  donnant 2 ruptures de pentes

#### Commentaire :

L’estimation régularisée avec pénalisation  $l_0$  sur le vecteur des pentes a permis d’atténuer le biais de régularisation lié à la norme 1 et de donner des estimées de  $\mathbf{R}$  représentatives. En effet, pour des valeurs de  $\lambda$  donnant le même nombre de ruptures de pentes les estimées sont identiques. De plus, le temps de calcul nécessaire à l’obtention d’une telle parcimonie ne dépasse pas les 6 minutes comme indiqué sur la Table 9. Pour des données de taille plus importante, le temps de calcul sera plus grand, cependant puisque nous travaillons sur des données journalières nous pouvons nous permettre d’effectuer une estimation du coefficient de reproduction avec un programme qui prend quelques heures si cela nous permettra d’obtenir une estimée continue dans le temps, non bruitée et fidèle au modèle épidémiologique.

## 5 Conclusions et perspectives

### 5.1 Conclusions

L'approche régularisée avec une pénalisation  $l_0$  nous a permis de remédier aux limites des travaux d'estimation du coefficient de reproduction avec une pénalisation  $l_1$  que nous avons présenté dans l'état de l'art. En effet, nous avons pu élaborer un problème d'optimisation permettant en théorie d'obtenir de la parcimonie exacte sur le vecteur des pentes associé au coefficient de reproduction mais aussi d'atténuer le biais de régularisation lié à la norme 1. Nous avons aussi implémenté un prototype de l'algorithme Branch-and-Bound en python afin de résoudre notre problème  $l_0$ . Cependant, le temps d'exécution de cet algorithme est une fonction croissante de la taille des données et une fonction décroissante du coefficient de régularisation  $\lambda$ . De plus, l'exécution du Branch-and-Bound passe par un calcul approximatif de la borne inférieure surtout pour des niveaux de régularisation assez faible. Cela dit obtenir quelques ruptures de pentes avec nos données d'origine prend quelques heures et malheureusement nous n'avons pas pu exploré en profondeur les résultats de l'estimation avec ces données. Néanmoins, nous avons vérifié le fonctionnement de notre algorithme en tronquant nos données et en le confrontant à une méthode de résolution systématique et donc exacte. Nous avons aussi mis en lumière les améliorations apportées par l'estimation avec une pénalisation  $l_0$ .

Les approximations effectuées pour le calcul de la borne inférieure sont principalement dues à la nature de la divergence de Kullback-Leibler (3) qui peut prendre la valeur  $+\infty$  mai aussi à l'application du seuillage doux pour le calcul des itérés des estimés. Pour obtenir des valeurs exactes nous pouvons reprendre à zéro notre étude en utilisant comme terme de fidélité au modèle épidémiologique la divergence de kullback-Leibler étendue (31). En effet, à l'instar de  $D_{KL}$ ,  $D_{KL}^{etendue}$  vérifie les hypothèses de convexité, semi-continuité inférieure et coercivité. De plus, elle est à valeur dans  $\mathbb{R}$  et donc ne prend jamais la valeur  $+\infty$ .

## **6 Annexes**

### **6.1 Annexe 1**

Ci-dessous un lien github vers le code python associé à l'implémentation du problème à l'analyse présenté dans l'état de l'art :

<https://github.com/Anasscobra/code-python-stage-analyse>

### **6.2 Annexe 2**

Ci-dessous un lien github vers le code python associé à l'implémentation du problème à la synthèse :

<https://github.com/Anasscobra/code-python-stage-synthese>

### **6.3 Annexe 3**

Ci-dessous un lien github vers le code python associé à l'implémentation du problème  $l_0$  :

<https://github.com/Anasscobra/code-python-stage-l0>

## 7 Bibliographie

### Références

- [1] Patrice Abry, Nelly Pustelnik, Stéphane Roux, Pablo Jensen, Patrick Flandrin, Rémi Gribonval, Charles-Gérard Lucas, Éric Guichard, Pierre Borgnat, and Nicolas Garnier. Spatial and temporal regularization to estimate covid-19 reproduction number  $r(t)$  : Promoting piecewise smoothness via convex optimization. *Plos one*, 15(8) :e0237901, 2020.
- [2] Dimitris Bertsimas and Romy Shioda. Algorithm for cardinality-constrained quadratic optimization. *Computational Optimization and Applications*, 43(1) :1–22, 2009.
- [3] Antonin Chambolle and Thomas Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of mathematical imaging and vision*, 40 :120–145, 2011.
- [4] P. L. Combettes and H. H. Bauschke. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. CMS books in mathematics. Springer Science + Business Media, New York, NY, 1. Aufl. edition, 2011.
- [5] Anne Cori, Neil M Ferguson, Christophe Fraser, and Simon Cauchemez. A new framework and software to estimate time-varying reproduction numbers during epidemics. *American journal of epidemiology*, 178(9) :1505–1512, 2013.
- [6] Juliana Du, Barbara Pascal, and Patrice Abry. Compared performance of covid19 reproduction number estimators based on realistic synthetic data. 2023.
- [7] Ramzi Ben Mhenni. *Méthodes de programmation en nombres mixtes pour l’optimisation parcimonieuse en traitement du signal*. PhD thesis, École Centrale de Nantes (ECN), 2020.
- [8] Barbara Pascal, Patrice Abry, Nelly Pustelnik, Stéphane Roux, Rémi Gribonval, and Patrick Flandrin. Nonsmooth convex optimization to estimate the covid-19 reproduction number space-time evolution with robustness against low quality data. *IEEE Transactions on Signal Processing*, 70 :2859–2868, 2022.