

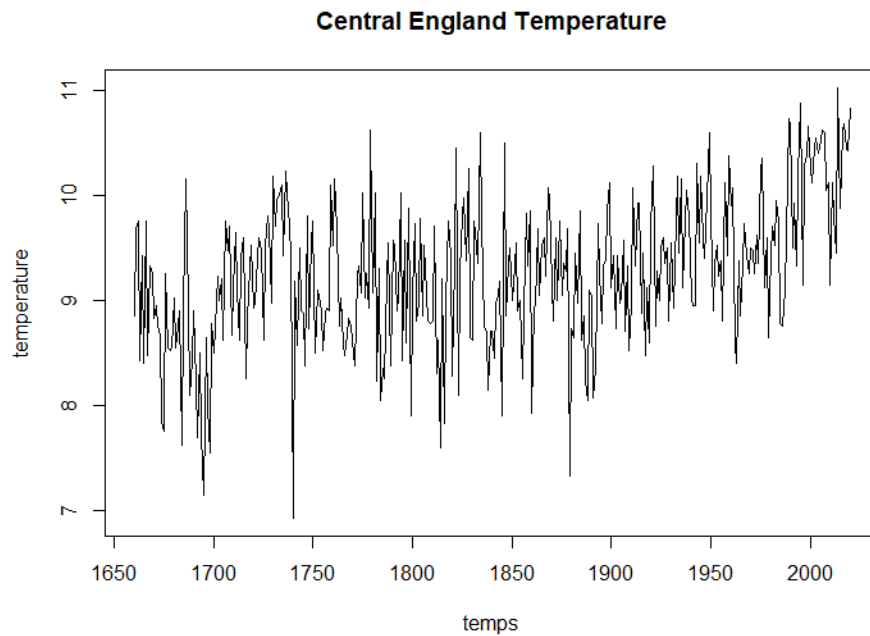
Faut-il croire au réchauffement climatique?

AMINATA NDIAYE

FANIRIANA RAKOTO ENDOR

RAPHAËL MALAK

Mai 2022



Sous la direction de Madalina Olteanu

Remerciements

Nous tenons à remercier notre encadrante Mme Madalina Olteanu, pour sa disponibilité, sa pédagogie et pour nous avoir aidés et orientés dans ce travail.

Nous la remercions également pour ce sujet qui nécessite divers outils mathématiques pour être traité. Nous avons pu mettre en pratique de nombreuses compétences théoriques que nous avons acquises durant cette première année de master.

Pour finir, nous souhaitons remercier l'ensemble du corps professoral du master 1 de mathématiques de l'université Paris-Dauphine pour leurs enseignements.

Table des matières

1	Introduction	5
1.1	Présentation du jeu de données	5
1.1.1	Origine, méthode de calcul et incertitude sur les données	5
1.1.2	Visualisation des données et hypothèses	6
1.2	Présentation des objectifs	9
2	Vocabulaire et notations	10
2.1	Série temporelle et notations	10
2.2	Point de rupture	10
2.3	Fonction de coût	10
2.3.1	Définition	10
2.3.2	Exemples et choix de fonctions de coût	11
2.3.3	Lien entre fonction de coût et estimateur du maximum de vraisemblance	11
2.4	Pénalité	12
2.4.1	Définition	12
2.4.2	Le BIC : origine et adéquation à notre problème	13
3	Différents modèles étudiés	16
3.1	Rupture dans la moyenne	16
3.1.1	Présentation du modèle	16
3.1.2	Estimateur des paramètres du modèle	16
3.2	Rupture dans la tendance	17
3.2.1	Présentation du modèle	17
3.2.2	Estimateur des paramètres du modèle	17
3.3	Le modèle de rupture en moyenne avec bruit AR(1)	19
3.3.1	Présentation du modèle	19
3.3.2	Estimateur des paramètres du modèle	19
3.4	Le modèle de rupture dans la tendance avec bruit AR(1)	21
3.4.1	Présentation du modèle	21
3.4.2	Estimateur des paramètres du modèle	21
4	Détection de points de rupture	24
4.1	Algorithme de programmation dynamique	24

4.1.1	Principe	24
4.1.2	Implémentation sous R	24
4.1.3	Complexité	25
4.1.4	Applications numériques et Résultats	26
4.2	Algorithme PELT	29
4.2.1	Principe	29
4.2.2	Implémentation sous R	30
4.2.3	Complexité	31
4.2.4	Applications numériques et Résultats	32
4.3	Comparaison de l'algorithme de programmation dynamique et de l'algorithme PELT	34
4.3.1	Complexité	34
4.3.2	Simulations	36
	Conclusion	38
	Bibliographie	39

1 Introduction

1.1 Présentation du jeu de données

1.1.1 Origine, méthode de calcul et incertitude sur les données

La base de données que nous allons étudier est la CET (Central England Temperature). Il s'agit de la plus grande série de températures mesurées au monde. Elles ont été enregistrées au centre de l'Angleterre, plus précisément dans la zone triangulaire formée par les villes de Londres dans le sud-est, Bristol dans le sud-ouest et Lancashire dans le nord. La série mensuelle débute en 1659 et est toujours enregistrée à ce jour. La série quotidienne, elle, n'a débuté qu'en 1772.

Gordon Manley, un climatologue et géographe britannique, a compilé une grande partie de la série mensuelle correspondant à la période allant de 1659 à 1973. Ces données ont été mises à jour jusqu'en 1991 par Parker, Legg et Folland. De nos jours, la série est mise à jour régulièrement par le Met Office, la section de surveillance des données du Hadley Centre.

Les stations utilisées pour compiler la CET sont actuellement Rothamsted (Hertfordshire), Pershore (Worcestershire) et Stonyhurst (Lancashire). Cependant, les stations utilisées ont changé au fil des ans. Après chaque modification, les données sont ajustées pour s'assurer de la cohérence avec la série historique. Des travaux récents (Parker et Horton, 2005) ont permis le calcul d'intervalles d'erreurs pour les moyennes annuelles depuis 1877.

Depuis 1974, les données ont été ajustées de 0,1 à 0,3 degré celsius pour tenir compte du réchauffement urbain.

En 2006, Croxton Hubert et Collinson comparent les températures mensuelles de stations situées un peu partout au Royaume-Uni et trouvent qu'elles sont fortement corrélées avec la CET, donc son applicabilité s'étend au-delà du centre de l'Angleterre.

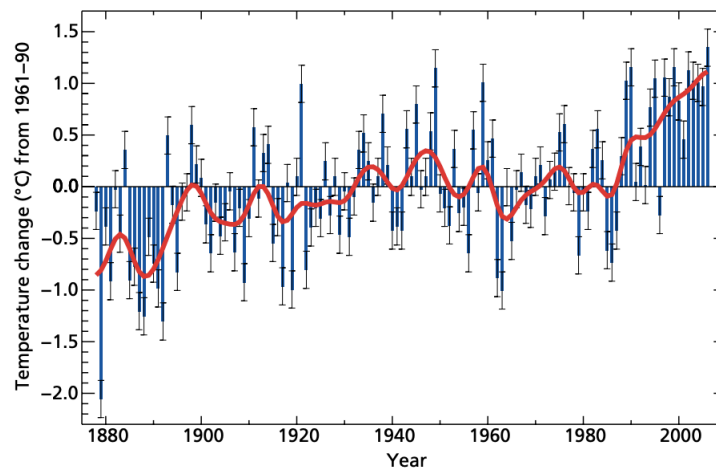


FIGURE 1 – Évolution des valeurs annuelles de la CET (barres bleues) de 1877 à 2006 par rapport à la moyenne sur la période de référence 1961-90 période (environ 9,5 °C). Les barres d'erreur entourent la plage(niveau de confiance de 95 %). La ligne rouge met l'accent sur les variations décennales. (La source : Met Office Handley Centre)

1.1.2 Visualisation des données et hypothèses

Ainsi la CET est une série de températures annuelles. Nous pouvons donc l'assimiler à une série temporelle. Regardons sa représentation graphique.

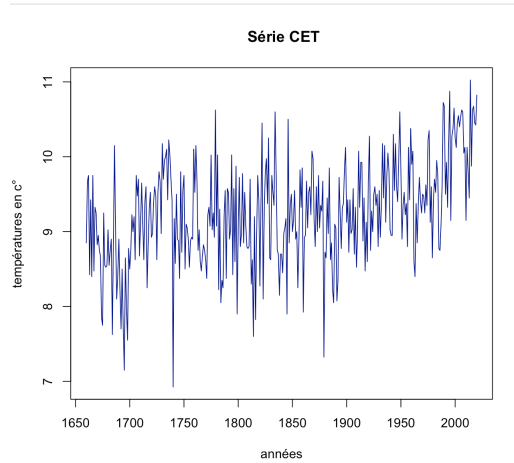


FIGURE 2 – Représentation graphique de la CET

Il est difficile de voir les particularités de la série en observant le graphe de la série brute. Nous allons donc observer le graphe lissé obtenu en utilisant, non pas les températures par années, mais les moyennes glissantes sur 5 ans. Nous obtenons le graphe suivant :

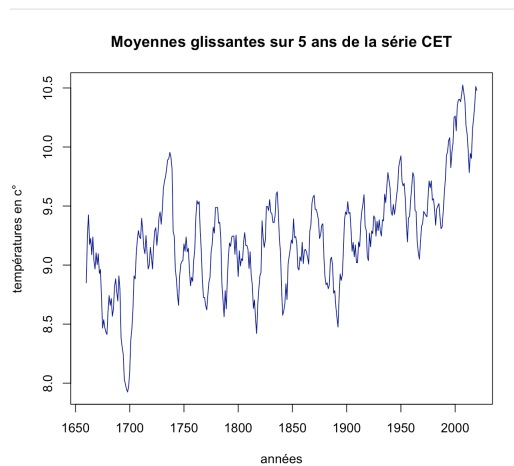


FIGURE 3 – Représentation graphique de la CET (moyennes glissantes sur 5 ans)

Grâce à cette nouvelle représentation, nous voyons plus aisément des plateaux de températures, certaines tendances se dessinent.

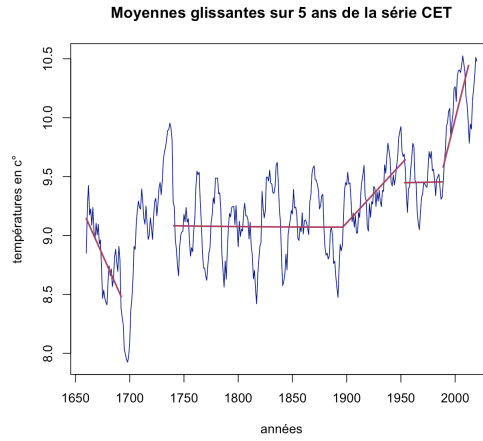


FIGURE 4 – Représentation graphique de la CET (moyennes glissantes sur 5 ans). Tendances dessinées à la main.

Nous constatons que notre jeu de données peut être segmenté en plusieurs séries avec des caractéristiques différentes, une moyenne différente ou une tendance d'évolution des températures différente. Ce constat va nous mener vers deux modélisations différentes de la CET : le modèle de rupture dans la moyenne et le modèle de rupture dans la tendance.

Comme nous l'avons dit précédemment, nous assimilons la CET à une série temporelle. Nous avons donc cherché à savoir s'il y avait une forte autocorrélation entre les données. Pour cela nous avons représenté le diagramme d'autocorrélation du jeu de données grâce à la fonction `acf` de R.

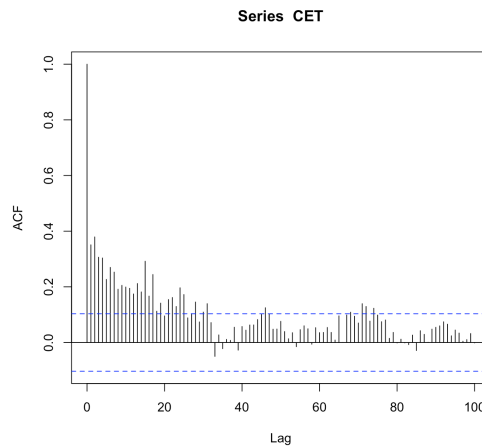


FIGURE 5 – Représentation graphique de la fonction d'autocorrélation de la CET

Nous pouvons constater que pour des années relativement proches les températures sont positivement corrélées. La fonction d'autocorrélation nous fournit des informations limitées. En effet, nous ne pouvons

pas connaître à partir de ce diagramme le niveau de corrélation au passé de la série. Pour cela, il faut s'intéresser à la fonction d'autocorrélation partielle.

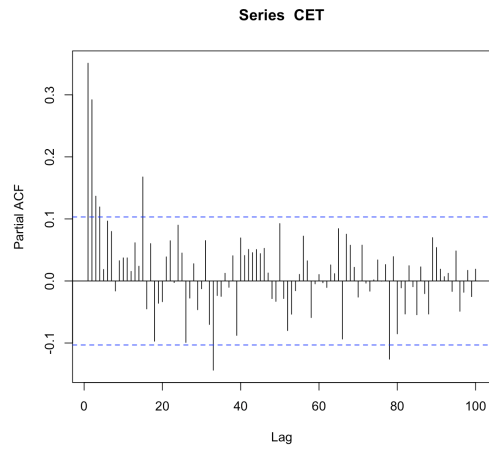


FIGURE 6 – Représentation graphique de la fonction d'autocorrélation partielle de la CET

Sur ce diagramme nous constatons que la dépendance au passé de la série est d'ordre assez faible. La température d'une année ne dépend, de façon significative, que de la température de l'année précédente. Nous allons nous servir de cette propriété de notre série pour enrichir nos modèles en ajoutant une composante autorégressive d'ordre 1.

1.2 Présentation des objectifs

L'étude d'une série statistique passe souvent par la connaissance de ses propriétés mais aussi par la détection des moments où ces dernières changent. Ainsi, La détection de points de rupture est un outil important d'analyse de séries. Elle présente un intérêt dans différents champs d'application comme la finance, l'océanographie, la biologie, et plus particulièrement dans notre cas, la climatologie.

Toute la difficulté dans la détection de points de rupture multiples est d'identifier la localisation et le nombre optimal de ces derniers étant donné que le champ des solutions possibles s'élargit considérablement avec la longueur du jeu de données.

Dans le cadre de ce mémoire, nous chercherons dans un premier temps à modéliser la série climatique qui nous a été proposée ; puis dans un second temps, pour chaque modèle, nous chercherons les points de ruptures optimaux qui permettent aux différents modèles de se rapprocher au plus du jeu de données.

Pour effectuer ce travail, nous nous aiderons des outils de statistique et d'analyse mathématique ainsi que du principe de la programmation dynamique. Une grande partie du travail reposera sur l'implémentation des algorithmes de détection de ruptures, à cet effet nous avons choisi d'utiliser le langage de programmation R.

2 Vocabulaire et notations

2.1 Série temporelle et notations

Nous considérons l'espace probabilisé $(\Omega, \mathcal{F}, \mathbb{P})$. Comme nous manipulons des températures qui appartiennent à \mathbb{R} , posons $\Omega = \mathbb{R}$ et $\mathcal{F} = \mathcal{B}(\mathbb{R})$, la tribu borélienne.

Nous assimilons la CET à une série temporelle, définissons ce terme. Pour cela, commençons par définir un processus stochastique.

Définition 2.1. *Un processus stochastique $\{X_t : t \in \mathbb{T}\}$ défini sur un espace d'indice \mathbb{T} est une collection de variables aléatoires définies sur le même espace de probabilité $(\Omega, \mathcal{F}, \mathbb{P})$.*

Définition 2.2. *Une série temporelle est un processus stochastique dont l'espace d'indice \mathbb{T} est soit \mathbb{N} , \mathbb{Z} , $[0, \infty[$ voir \mathbb{R}*

Dans notre cas, les températures sont annuelles, l'espace d'indice \mathbb{T} est donc \mathbb{N} . Nous introduisons les notations suivantes que nous utiliserons par la suite.

Soit $x = (x_1, \dots, x_n)$ une observation du processus $X = (X_1, \dots, X_n)$, pour tout $\tau \in \llbracket 1, n \rrbracket$ on note $x_{1:\tau}$ la sous série (x_1, \dots, x_τ) . De même pour tout $\tau_1, \tau_2 \in \llbracket 1, T \rrbracket$ tel que $\tau_1 < \tau_2$, $x_{\tau_1:\tau_2} = (x_{\tau_1}, \dots, x_{\tau_2})$.

2.2 Point de rupture

La première notion qui va nous intéresser pour l'étude du comportement de notre série est celle de point de rupture.

Définition 2.3. *Un point de rupture est un moment où les propriétés statistiques de la série que l'on observe changent. Soit une séquence ordonnée de données $x_{1:n}$, on dit qu'il y a un point de rupture dans la série s'il existe $\tau \in \llbracket 1, \dots, n-1 \rrbracket$ tel que les propriétés statistiques de $x_{1:\tau}$ et $x_{\tau+1:n}$ soient différentes.*

Dans le cas de points de rupture multiples, soit $k \in \llbracket 1, \dots, n-1 \rrbracket$ le nombre de points de rupture et $\tau_{1:k} = (\tau_1, \dots, \tau_k)$, chaque point de rupture est un entier entre 1 et $n-1$ inclus. On définit $\tau_0 = 0$ et $\tau_{k+1} = n$, on suppose que les points de rupture sont ordonnés tel que $\tau_i < \tau_j$ si et seulement si $i < j$. Ainsi les k points de rupture divisent le jeu de données en $k+1$ segments, le i ème segment correspondant à $x_{(\tau_{i-1}+1):\tau_i}$.

2.3 Fonction de coût

La fonction de coût nous permet de quantifier l'erreur entre les données empiriques et les estimations fournies par nos modèles. Ainsi, nous avons pour objectif d'avoir le modèle avec le coût le plus faible possible. La fonction de coût doit donc être minimisée.

2.3.1 Définition

Définition 2.4. *Soit y, \hat{y} dans \mathbb{R}^m . Une fonction de coût est une fonction mesurable \mathcal{C} telle que :*

$$\mathcal{C} : \left| \begin{array}{ccc} \mathbb{R}^m \times \mathbb{R}^m & \longrightarrow & \mathbb{R} \\ (y, \hat{y}) & \longmapsto & \mathcal{C}(y, \hat{y}) \end{array} \right.$$

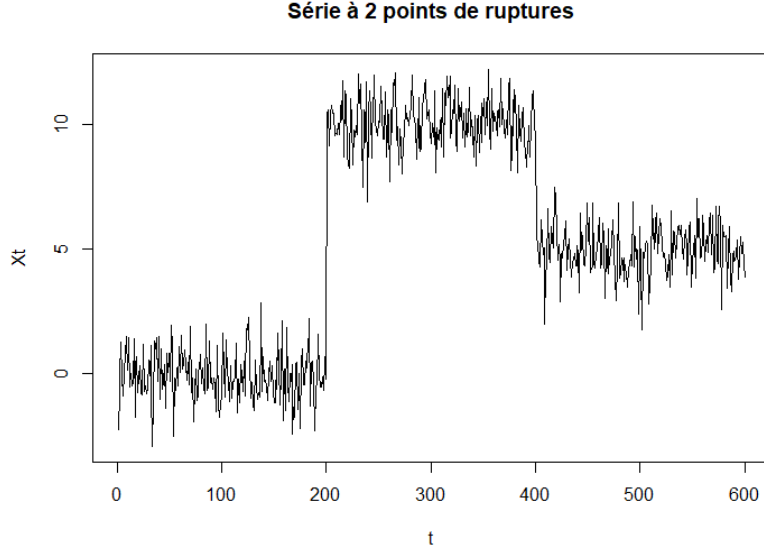


FIGURE 7 – Série de taille $n = 600$ avec deux points de rupture, $\tau_1 = 200$ et $\tau_2 = 400$. Les sous séries $x_{0:\tau_1}$, $x_{\tau_1:\tau_2}$, $x_{\tau_2:n}$ ont des moyennes différentes, respectivement 0, 10, 5.

Remarque 2.1. Dans un cadre paramétrique où \hat{y} peut être exprimé en fonction de y et d'un paramètre $\theta \in \mathbb{R}^d$, la fonction de coût peut se définir comme suit :

$$\mathcal{C} : \begin{cases} \mathbb{R}^m \times \mathbb{R}^d & \longrightarrow \mathbb{R} \\ (y, \theta) & \longmapsto \mathcal{C}(y, \theta) \end{cases}$$

Nous utiliserons par la suite la notation suivante : $\mathcal{C}(y) = \min_{\theta} \mathcal{C}(y, \theta)$

2.3.2 Exemples et choix de fonctions de coût

Définition 2.5 (Fonction de coût quadratique). Il s'agit de la fonction de coût la plus utilisée pour les problèmes de régression.

$$\mathcal{C} : \begin{cases} \mathbb{R}^m \times \mathbb{R}^m & \longrightarrow \mathbb{R} \\ (y, \hat{y}) & \longmapsto \sum_{t=1}^m (y_t - \hat{y}_t)^2 \end{cases}$$

Cette fonction de coût nous permet de mesurer la distance, pour chaque donnée, entre la valeur prédite par notre modèle et sa valeur réelle. De plus, l'affinité qui existe entre la fonction de coût quadratique et la log-vraisemblance d'un vecteur gaussien fait de cette fonction de coût le choix idoine pour nos modèles.

2.3.3 Lien entre fonction de coût et estimateur du maximum de vraisemblance

Nous établissons dans cette partie un lien entre la fonction de coût d'un modèle et son maximum de vraisemblance. On considère la série $y = (y_t)_{t \leq n}$ à K points de rupture qui s'écrit comme :

$$y_t = \begin{cases} \mu_0 + z_t, & 1 \leq t \leq \tau_1 \\ \mu_1 + z_t, & \tau_1 + 1 \leq t \leq \tau_2 \\ \vdots \\ \mu_K + z_t, & \tau_K + 1 \leq t \leq n \end{cases}$$

Avec $(z_t)_t \stackrel{iid}{\sim} \mathcal{N}(0, \sigma^2)$.

La log-vraisemblance de la série y , notée \mathcal{L}_n , vaut :

$$\mathcal{L}_n = \sum_{k=1}^K \sum_{t=\tau_k+1}^{\tau_{k+1}} \ln \left(\frac{1}{\sqrt{2\pi\sigma_k^2}} \exp \left(-\frac{(y_t - \mu_k)^2}{2\sigma^2} \right) \right)$$

En notant, \mathcal{C} la fonction de coût quadratique, cela équivaut à :

$$\begin{aligned} \mathcal{L}_n &= \ln \left(\frac{1}{\sqrt{2\pi\sigma^2}} \right) - \frac{1}{2\sigma^2} \sum_{k=1}^K \sum_{t=\tau_k+1}^{\tau_{k+1}} (y_t - \mu_k)^2 \\ &= \ln \left(\frac{1}{\sqrt{2\pi\sigma^2}} \right) - \frac{1}{2\sigma^2} \mathcal{C}(y, \mu) \text{ (avec } \mu = (\mu_1, \dots, \mu_K)) \end{aligned}$$

Ainsi, maximiser la log-vraisemblance revient à minimiser la fonction de coût conditionnellement au fait que nous connaissons σ . Dans les parties suivantes, pour estimer les paramètres des modèles étudiés sur le segment $\{\tau_k + 1, \dots, \tau_{k+1}\}$, nous allons minimiser la fonction de coût \mathcal{C} pour $\mu_k \in \mathbb{R}$ qui sera dépendant du modèle.

2.4 Pénalité

Lorsque le nombre de points de rupture K est inconnu, on introduit une pénalité $pen(\cdot)$ afin d'imposer une contrainte sur le nombre de points de rupture en fixant un nombre maximum de points de rupture K_{max} . En posant $\hat{y}^{(k)}$ le modèle à k points de rupture de la série y étudiée, le problème devient le suivant :

$$\min_{\substack{k \leq K_{max} \\ \tau_1 \leq \dots \leq \tau_k}} \mathcal{C}(y, \hat{y}^{(k)}) + pen(k) \quad (1)$$

2.4.1 Définition

Nous choisirons comme pénalité le critère d'information bayésien (BIC) dans le cadre du problème (1).

Définition 2.6. Soit $\hat{\mathcal{L}}$ la valeur maximale de l'estimateur du maximum de vraisemblance pour un modèle donné, $p \in \mathbb{N}$ le nombre de paramètres libres du problème et $n \in \mathbb{N}$ la longueur de la série. La pénalité BIC est définie comme suit :

$$BIC(p) = -2\ln(\hat{\mathcal{L}}) + p\ln(n)$$

2.4.2 Le BIC : origine et adéquation à notre problème

Dans cette partie nous présentons la construction du BIC. Nous nous appuyons sur l'article de E. Lebarbier et M-H. Tristan [3].

Nous disposons d'un n-échantillon $X = (X_1, \dots, X_n)$ de variables aléatoires indépendantes et identiquement distribuées de densité inconnue f . L'objectif est d'estimer f . Soit $\{M_1, \dots, M_m\}$ une collection finie de modèles. Un modèle M_i correspond à une densité g_{M_i} de paramètre θ_i de dimension p_i . On note Θ_i l'espace de dimension p_i auquel appartient θ_i . L'objectif est de choisir le meilleur modèle parmi cette collection de modèle.

Le critère BIC se place dans un contexte bayésien : θ_i et M_i sont vus comme des variables aléatoires et sont munis d'une distribution a priori :

- La distribution a priori sur M_i est noté $P(M_i)$.
- Pour un modèle M_i donné, la distribution a priori de θ_i est noté $P(\theta_i|M_i)$.

Grâce à cette approche nous pouvons prendre en compte des informations que peut détenir l'utilisateur, en donnant à certains modèles un poids plus important. Le BIC cherche à sélectionner le modèle M_i qui maximise la probabilité a posteriori $P(M_i|X)$:

$$M_{BIC} = \arg \max_{M_i} P(M_i|X)$$

Le BIC cherche à sélectionner le modèle M_i le plus vraisemblable aux données.

D'après la formule de Bayes, $P(M_i|X)$ s'écrit :

$$P(M_i|X) = \frac{P(X|M_i)P(M_i)}{P(X)}$$

Nous supposons dans la suite que la loi a priori des modèles de M_i est non informative, c'est-à-dire que :

$$P(M_1) = P(M_2) = \dots = P(M_m)$$

Ainsi aucun modèle n'est privilégié. Sous cette hypothèse, la recherche du meilleur modèle ne nécessite que le calcul de la distribution $P(M_i|X)$:

$$P(M_i|X) = \int_{\Theta_i} P(X, \theta_i|M_i) d\theta_i = \int_{\Theta_i} g_{M_i}(X, \theta_i) P(\theta_i|M_i) d\theta_i$$

où $g_{M_i}(X, \theta_i)$ est la vraisemblance correspondant au modèle M_i de paramètre θ_i :

$$g_{M_i}(X, \theta_i) = P(\theta_i|X, M_i)$$

Propriété 2.1 (Approximation de Laplace). *Soit une fonction $L : \mathbb{R}^d \rightarrow \mathbb{R}$ telle que L est deux fois différentiable sur \mathbb{R} et atteint un unique maximum sur \mathbb{R} en u^* . Alors,*

$$\int_{\mathbb{R}^d} e^{nL(u)} du = \int_{\mathbb{R}^d} e^{nL(u^*)} \left(\frac{2\pi}{n} \right)^{\frac{d}{2}} | -L''(u^*) |^{-\frac{1}{2}} + \mathcal{O}(n^{-1}).$$

Posons $g(\theta_i) = \log(g_{M_i}(X, \theta_i)P(\theta_i|M_i))$.

Nous appliquons l'approximation de Laplace à la fonction suivante :

$$L_n(\theta_i) = \frac{g(\theta_i)}{n} = \frac{1}{n} \sum_{k=1}^n \log(g_{M_i}(X_k, \theta_i)) + \frac{\log(P(\theta_i|M_i))}{n}$$

Nous notons :

- $\theta_i^* = \arg \max_{\theta_i \in \Theta_i} L_n(\theta_i)$,
- $A_{\theta_i^*}$ l'opposé de la matrice hessienne des dérivées secondes partielles de la fonction $L_n(\theta_i)$ en θ_i :

$$A_{\theta_i^*} = - \left[\frac{\partial^2 L_n(i)}{\partial \theta_i^j \partial \theta_i^l} \right]_{j,l} \Big|_{\theta_i = \theta_i^*},$$

où θ_i^j est la jème composante du vecteur des paramètres θ_i .

Nous obtenons :

$$P(X|M_i) = e^{g(\theta_i^*)} \left(\frac{2\pi}{n} \right) \frac{p_i}{2} |A_{\theta_i^*}|^{-\frac{1}{2}} + \mathcal{O}(n^{-1}),$$

d'où :

$$\log(P(X|M_i)) = \log(g_{M_i}(X, \theta_i^*)) + \log(P(\theta_i^*|M_i)) + \frac{p_i}{2} \log(2\pi) - \frac{1}{2} \log(|A_{\theta_i^*}|) + \mathcal{O}(n^{-1}) \quad (2)$$

Asymptotiquement θ_i^* peut être remplacé par l'estimateur du maximum de vraisemblance $\hat{\theta}_i$:

$$\hat{\theta}_i = \arg \max_{\theta_i \in \Theta_i} \frac{1}{n} g_{M_i}(X, \theta_i),$$

et $A_{\theta_i^*}$ par la matrice d'information de Fisher pour une observation que l'on va noter $I_{\hat{\theta}_i}$ définie par :

$$I_{\hat{\theta}_i} = -\mathbb{E} \left(\left[\frac{\partial^2 \log(g_{M_i}(X_1, \theta_i))}{\partial \theta_i^j \partial \theta_i^l} \right]_{j,l} \Big|_{\theta_i = \hat{\theta}_i} \right).$$

Lorsque n est grand $\log(g_{M_i}(X, \theta_i)P(\theta_i|M_i))$ se comporte comme $\log(g_{M_i}(X, \theta_i))$ qui croît avec n tandis que $\log(P(\theta_i^*|M_i))$ reste constant. Nous admettons qu'en remplaçant θ_i^* par $\hat{\theta}_i$ et $A_{\theta_i^*}$ par $I_{\hat{\theta}_i}$ dans (2), nous introduisons un terme d'erreur en $n^{-\frac{1}{2}}$. Nous obtenons :

$$\log(P(X|M_i)) = \overbrace{\log(g_{M_i}(X, \hat{\theta}_i)) - \frac{p_i}{2} \log(n)}^{\text{tend vers } -\infty \text{ avec } n} + \overbrace{\log(P(\hat{\theta}_i|M_i)) + \frac{p_i}{2} \log(2\pi) - \frac{1}{2} \log(|I_{\hat{\theta}_i}|) + \mathcal{O}(n^{-\frac{1}{2}})}^{\text{reste borné : } \mathcal{O}(1)}.$$

En négligeant les termes d'erreurs $\mathcal{O}(1)$ et $\mathcal{O}(n^{-\frac{1}{2}})$, nous obtenons :

$$\log(P(X|M_i)) = \log(g_{M_i}(X, \hat{\theta}_i)) - \frac{p_i}{2} \log(n)$$

.

Le critère BIC est issu de cette approximation. Pour le modèle M_i , il correspond à l'approximation de $-2 \log(P(X|M_i))$ et est donc défini par :

$$BIC_i^* = -2 \log(g_{M_i}(X, \hat{\theta}_i)) + p_i \log(n).$$

Le modèle sélectionné par le critère BIC est :

$$M_{BIC} = \arg \min_{M_i} BIC_i$$

Etant donné la relation étroite entre la log-vraisemblance de variables aléatoires gaussiennes et la fonction de coût quadratique, nous allons par la suite utiliser le critère simplifié :

$$M_{BIC^*} = \arg \min_{M_i} BIC_i^*$$

Où BIC_i^* est :

$$BIC_i = \mathcal{C}(M_i, X, \hat{\theta}_i) + p_i \log(n).$$

avec $\mathcal{C}(M_i, X, \hat{\theta}_i)$ est le coût du modèle M_i de paramètre optimal $\hat{\theta}_i$.

Remarquons cependant que pour appliquer la pénalité du BIC, nous devons disposer de données indépendantes et identiquement distribuées. Dans les modèles que nous allons vous présenter, ces hypothèses ne seront pas nécessairement vérifiées.

3 Différents modèles étudiés

Pour être capable de conjecturer certains comportements de la CET - par exemple un réchauffement de la température au 20^{ème} siècle - nous allons poser plusieurs schémas (appelés aussi modèles). Un modèle peut être considéré comme une estimation statistique du mécanisme ayant généré nos observations. Nous proposons donc dans ce mémoire 4 différents modèles pour analyser sous différents angles la série temporelle.

Nous supposons dans cette section que le nombre de points de rupture est connu à l'avance, nous le notons K . Nous noterons τ_1, \dots, τ_K les K points de rupture.

Définissons $\llbracket 0, \tau_1 \rrbracket$ le segment 0, $\llbracket \tau_k + 1, \tau_{k+1} \rrbracket, k = 1, \dots, K - 1$ le segment k et $\llbracket \tau_K + 1, n \rrbracket$ le segment K . Posons $(z_t)_t$ un bruit blanc iid de variance supposée finie tel que $z_t \sim \mathcal{N}(0, \sigma^2)$.

Chacun des 4 modèles est défini par morceaux sur les $K + 1$ segments obtenus. Nous présentons ci-après nos modèles du plus simple au plus complexe.

3.1 Rupture dans la moyenne

3.1.1 Présentation du modèle

Notre première modélisation nécessite seulement de connaître la notion de moyenne empirique.

Définition 3.1. (*Moyenne empirique*) Soit $X = (X_1, \dots, X_n)$ un n -échantillon suivant une loi \mathcal{L} . Supposons que \mathcal{L} soit d'espérance finie, c'est-à-dire $\mathbb{E}[Y] < +\infty$ pour $Y \sim \mathcal{L}$. La **moyenne empirique** (notée $\bar{\mu}$) est un estimateur fortement consistant de l'espérance de \mathcal{L} :

$$\bar{\mu} = \frac{1}{n} \sum_{i=1}^n X_i$$

Le modèle de rupture dans la moyenne se résume simplement à un calcul des moyennes empiriques par segment. Il est de la forme :

$$y_t = \begin{cases} \mu_0 + z_t, & 1 \leq t \leq \tau_1 \\ \mu_1 + z_t, & \tau_1 + 1 \leq t \leq \tau_2 \\ \vdots \\ \mu_K + z_t, & \tau_K + 1 \leq t \leq n \end{cases}$$

Le modèle étant maintenant présenté, il serait bon de se demander si celui-ci est pertinent. L'idée derrière ce modèle est que pour chaque segment, où la variance de la série est plus faible qu'au niveau des points de rupture, nous approchons la moyenne de la CET. Donc plus la taille des segments est grande, autrement dit plus il y a d'observations entre 2 points de rupture, et plus l'estimation de ces moyennes est précise (par la loi des grands nombres). La finalité est de "lisser" cette série afin d'en déceler certaines caractéristiques.

3.1.2 Estimateur des paramètres du modèle

Les estimateurs de $\mu = (\mu_0, \dots, \mu_K)$ sont déterminés en minimisant la fonction de coût quadratique. Dans notre modèle de rupture dans la moyenne, pour un segment k donné, la fonction de coût est de la

forme :

$$\mathcal{C}(y, \mu_k) = \sum_{t=\tau_k+1}^{\tau_{k+1}} (y_t - \mu_k)^2$$

En dérivant cette fonction par rapport à μ_k , nous obtenons :

$$\begin{aligned} \frac{\partial \mathcal{C}(y, \mu_k)}{\partial \mu_k} &= 2 \sum_{t=\tau_k+1}^{\tau_{k+1}} (\mu_k - y_t) \\ &= 2 \sum_{t=\tau_k+1}^{\tau_{k+1}} \mu_k - 2 \sum_{t=\tau_k+1}^{\tau_{k+1}} y_t \\ &= 2(\tau_{k+1} - \tau_k) \mu_k - 2 \sum_{t=\tau_k+1}^{\tau_{k+1}} y_t \end{aligned}$$

Cette dérivée s'annule en $\mu_k^* = \frac{1}{\tau_{k+1} - \tau_k} \sum_{t=\tau_k+1}^{\tau_{k+1}} y_t =: \bar{Y}_i$. De plus :

$$\frac{\partial^2 \mathcal{C}(y, \mu_k)}{\partial^2 \mu_k} = \tau_{k+1} - \tau_k > 0$$

L'inégalité découle de la stricte croissance des points de rupture (2 points de rupture ne peuvent pas être détectés au même moment). La fonction de coût est alors convexe sur le segment k , ce qui montre que μ_k^* est un minimum global de $\mathcal{C}(y, \mu_k)$ sur $[\tau_k + 1, \tau_{k+1}]$.

Remarque 3.1. μ_k^* est en fait la moyenne empirique de la série temporelle sur le segment k .

3.2 Rupture dans la tendance

3.2.1 Présentation du modèle

Sur le segment k , la série sera approchée par une droite qui lui est propre de la forme $\mu_k + \beta_k t$, où μ_k est appelé **intercept** (ou ordonnée à l'origine) et β_k est la **tendance** de la droite (ou le coefficient directeur). Le modèle de rupture dans la tendance se définit alors comme suit :

$$y_t = \begin{cases} \mu_0 + \beta_0 t + z_t, & 1 \leq t \leq \tau_1 \\ \mu_1 + \beta_1 t + z_t, & \tau_1 + 1 \leq t \leq \tau_2 \\ \vdots \\ \mu_K + \beta_K t + z_t, & \tau_K + 1 \leq t \leq n \end{cases}$$

3.2.2 Estimateur des paramètres du modèle

Comme dans la sous-partie 3.1.2, nous allons chercher à minimiser la fonction de coût quadratique associée à notre modèle afin de trouver les paramètres. Pour un segment k donné, cette fonction est de la forme :

$$\mathcal{C}(y, (\mu_k, \beta_k)) = \sum_{t=\tau_k+1}^{\tau_{k+1}} (y_t - \mu_k - \beta_k t)^2$$

Il s'agit donc ici de dériver \mathcal{C} par rapport à chacun des paramètres, déterminer les minimiseurs μ_k^* et β_k^* et ensuite de résoudre l'équation à 2 variables et 2 inconnues obtenue.

D'une part :

$$\begin{aligned}\frac{\partial \mathcal{C}(y, (\mu_k, \beta_k))}{\partial \mu_k} &= \sum_{t=\tau_k+1}^{\tau_{k+1}} 2(\mu_k + \beta_k t - y_t) \\ &= 2(\tau_{k+1} - \tau_k)\mu_k + 2\beta_k \sum_{t=\tau_k+1}^{\tau_{k+1}} t - 2 \sum_{t=\tau_k+1}^{\tau_{k+1}} y_t \\ &= 2(\tau_{k+1} - \tau_k)\mu_k + \beta_k(\tau_k + \tau_{k+1} + 1)(\tau_{k+1} - \tau_k) - 2 \sum_{t=\tau_k+1}^{\tau_{k+1}} y_t\end{aligned}$$

Nous cherchons à annuler la dérivée afin de trouver un extremum de la fonction de coût quadratique associée à notre modèle. Nous obtenons :

$$\mu_k^* = \bar{Y}_i - \beta_k \frac{(\tau_k + \tau_{k+1} + 1)}{2}$$

De plus :

$$\frac{\partial^2 \mathcal{C}(y, (\mu_k, \beta_k))}{\partial^2 \mu_k} = 2(\tau_{k+1} - \tau_k) > 0$$

La fonction de coût est alors convexe sur le segment k, ce qui montre que μ_k^* est un minimum global de $\mathcal{C}(y, (\mu_k, \beta_k))$ sur $[\tau_k + 1, \tau_{k+1}]$.

En notant $\bar{t}_i = \frac{\tau_k + \tau_{k+1} + 1}{2}$, nous avons de la même manière :

$$\begin{aligned}\frac{\partial \mathcal{C}(y, (\mu_k, \beta_k))}{\partial \beta_k} &= \sum_{t=\tau_k+1}^{\tau_{k+1}} 2t(\mu_k + \beta_k t - y_t) \\ &= 2\bar{t}_i(\tau_{k+1} - \tau_k)\mu_k + 2\beta_k \sum_{t=\tau_k+1}^{\tau_{k+1}} t^2 - 2 \sum_{t=\tau_k+1}^{\tau_{k+1}} ty_t\end{aligned}$$

Cette dérivée s'annule en $\beta_k^* = \frac{\sum_{t=\tau_k+1}^{\tau_{k+1}} ty_t - \bar{t}_i(\tau_{k+1} - \tau_k)\hat{\mu}_k}{\sum_{t=\tau_k+1}^{\tau_{k+1}} t^2}$

De plus :

$$\frac{\partial^2 \mathcal{C}(y, (\mu_k, \beta_k))}{\partial^2 \beta_k} = 2 \sum_{t=\tau_k+1}^{\tau_{k+1}} t^2 > 0$$

La fonction de coût est alors convexe sur le segment k, ce qui montre que β_k^* est un minimum global de $\mathcal{C}(y, (\mu_k, \beta_k))$ sur $[\tau_k + 1, \tau_{k+1}]$.

En insérant μ_k^* dans β_k^* , nous obtenons :

$$\beta_k^* = \frac{\sum_{t=\tau_k+1}^{\tau_{k+1}} ty_t - \bar{t}_i^2(\tau_{k+1} - \tau_k)\beta_k^* - \bar{t}_i(\tau_{k+1} - \tau_k)\bar{Y}_i}{\sum_{t=\tau_k+1}^{\tau_{k+1}} t^2}$$

D'où :

$$\beta_k^* = \frac{\sum_{t=\tau_k+1}^{\tau_{k+1}} ty_t - \bar{t}_i(\tau_{k+1} - \tau_k)\bar{Y}_i}{\sum_{t=\tau_k+1}^{\tau_{k+1}} t^2 - \bar{t}_i^2(\tau_{k+1} - \tau_k)}$$

Les calculs qui suivent ont pour but de simplifier l'expression de β_k^* :

$$\begin{aligned} \beta_k^* &= \frac{\sum_{t=\tau_k+1}^{\tau_{k+1}} ty_t - \sum_{t=\tau_k+1}^{\tau_{k+1}} \bar{t}_i \bar{Y}_i - \bar{t}_i(\tau_{k+1} - \tau_k)\bar{Y}_i + \bar{t}_i(\tau_{k+1} - \tau_k)\bar{Y}_i}{\sum_{t=\tau_k+1}^{\tau_{k+1}} t^2 - 2\bar{t}_i^2(\tau_{k+1} - \tau_k) + \bar{t}_i^2(\tau_{k+1} - \tau_k)} \\ &= \frac{\sum_{t=\tau_k+1}^{\tau_{k+1}} ty_t - \bar{t}_i \sum_{t=\tau_k+1}^{\tau_{k+1}} y_t - \bar{Y}_i \sum_{t=\tau_k+1}^{\tau_{k+1}} t + \sum_{t=\tau_k+1}^{\tau_{k+1}} \bar{t}_i \bar{Y}_i}{\sum_{t=\tau_k+1}^{\tau_{k+1}} t^2 - \sum_{t=\tau_k+1}^{\tau_{k+1}} 2\bar{t}_i t(\tau_{k+1} - \tau_k) + \bar{t}_i^2(\tau_{k+1} - \tau_k)} \\ &= \frac{\sum_{t=\tau_k+1}^{\tau_{k+1}} (y_t - \bar{Y}_i)(t - \bar{t}_i)}{\sum_{t=\tau_k+1}^{\tau_{k+1}} (t - \bar{t}_i)^2} \end{aligned}$$

Nos estimateurs optimaux sont finalement :

$$\begin{cases} \beta_k^* = \frac{\sum_{t=\tau_k+1}^{\tau_{k+1}} (y_t - \bar{Y}_i)(t - \bar{t}_i)}{\sum_{t=\tau_k+1}^{\tau_{k+1}} (t - \bar{t}_i)^2} \\ \mu_k^* = \bar{Y}_i - \beta_k^* \bar{t}_i \end{cases}$$

3.3 Le modèle de rupture en moyenne avec bruit AR(1)

3.3.1 Présentation du modèle

Sur le segment k la série sera modélisée par sa moyenne. Cette fois-ci, on considère un bruit auto-regressif $\varepsilon_t = \phi_k \varepsilon_{t-1} + z_t$ avec $z_t \sim \mathcal{N}(0, \sigma^2)$. La série s'écrit alors sous la forme :

$$y_t = \begin{cases} \mu_0 + \varepsilon_t, & 1 \leq t \leq \tau_1 \\ \mu_1 + \varepsilon_t, & \tau_1 + 1 \leq t \leq \tau_2 \\ \vdots \\ \mu_K + \varepsilon_t, & \tau_K + 1 \leq t \leq n \end{cases}$$

3.3.2 Estimateur des paramètres du modèle

Comme précédemment, nous cherchons à minimiser notre fonction de coût quadratique associée au modèle sur le segment k pour trouver les paramètres. Sur le segment k la série est de la forme $y_t = \mu_k + \varepsilon_t$, la fonction de coût s'obtient donc comme suit :

$$\begin{aligned}
y_t &= \mu_k + \varepsilon_t \\
&= \mu_k + \phi_k \varepsilon_{t-1} + z_t \\
&= \mu_k + \phi_k (y_{t-1} - \mu_k) + z_t \\
&= \phi_k y_{t-1} + \mu_k (1 - \phi_k) + z_t
\end{aligned}$$

On a ainsi $y_t - \phi_k y_{t-1} - \mu_k (1 - \phi_k) \sim \mathcal{N}(0, \sigma^2)$. En posant $\delta_k := \mu_k (1 - \phi_k)$ la fonction de coût quadratique sur le segment k , qu'on optimisera par rapport à ϕ_k et δ_k devient :

$$\mathcal{C}(y, (\phi_k, \delta_k)) = \sum_{t=\tau_k+1}^{\tau_{k+1}} (y_t - \phi_k y_{t-1} - \delta_k)^2 \quad (3)$$

Remarque 3.2. Notons que lorsque $t = \tau_k + 1$, $\varepsilon_{\tau_{k+1}} = \phi_k (y_{\tau_k} - \mu_{k-1}) + z_{\tau_{k+1}}$. Toutefois, nous considérerons l'approximation $\varepsilon_{\tau_{k+1}} = \phi_k (y_{\tau_k} - \mu_k) + z_{\tau_{k+1}}$ afin de simplifier nos calculs. Pour ces raisons, il s'agit d'une quasi-vraisemblance. Ceci reste vrai pour le modèle de rupture en tendance avec bruit $AR(1)$.

L'application $(\phi_k, \delta_k) \mapsto \mathcal{C}(y, (\phi_k, \delta_k))$ est de classe $\mathcal{C}^1(\mathbb{R}^2, \mathbb{R})$ et strictement convexe. Ainsi, annuler ses dérivées partielles nous donnera les uniques estimateurs δ_k^* et ϕ_k^* . D'une part, on a :

$$\frac{\partial \mathcal{C}(y, (\phi_k, \delta_k))}{\partial \phi_k} = \sum_{t=\tau_k+1}^{\tau_{k+1}} -2y_{t-1}(y_t - \phi_k y_{t-1} - \delta_k)$$

Qui s'annule lorsque :

$$\phi_k = \frac{\sum_{t=\tau_k+1}^{\tau_{k+1}} y_{t-1}(y_t - \delta_k)}{\sum_{t=\tau_k+1}^{\tau_{k+1}} y_{t-1}^2} \quad (4)$$

D'autre part :

$$\frac{\partial \mathcal{C}(y, (\phi_k, \delta_k))}{\partial \delta_k} = \sum_{t=\tau_k+1}^{\tau_{k+1}} -2(y_t - \phi_k y_{t-1} - \delta_k)$$

En posant $\Delta_k := \tau_{k+1} - \tau_k$, $\frac{\partial \mathcal{C}}{\partial \delta_k}$ s'annule lorsque :

$$\delta_k = \frac{\sum_{t=\tau_k+1}^{\tau_{k+1}} (y_t - \phi_k y_{t-1})}{\Delta_k} \quad (5)$$

En injectant l'expression de (5) dans (4) on obtient l'estimateur ϕ_k^* de ϕ_k :

$$\phi_k^* = \frac{\Delta_k \sum_{t=\tau_k+1}^{\tau_{k+1}} y_t y_{t-1} - \sum_{t=\tau_k+1}^{\tau_{k+1}} y_t \sum_{t=\tau_k+1}^{\tau_{k+1}} y_{t-1}}{\Delta_k \sum_{t=\tau_k+1}^{\tau_{k+1}} y_{t-1}^2 - (\sum_{t=\tau_k+1}^{\tau_{k+1}} y_{t-1})^2}$$

Et en injectant ϕ_k^* dans (5), on obtient l'estimateur δ_k^* de δ_k :

$$\delta_k^* = \frac{\sum_{t=\tau_k+1}^{\tau_{k+1}} (y_t - \phi_k^* y_{t-1})}{\Delta_k}$$

On obtient alors μ_k^* , l'estimateur de μ_k comme $\mu_k^* = \frac{\delta_k^*}{1-\phi_k^*}$.

Finalement, nos estimateurs sont :

$$\begin{cases} \phi_k^* = \frac{\Delta_k \sum_{t=\tau_k+1}^{\tau_{k+1}} y_t y_{t-1} - \sum_{t=\tau_k+1}^{\tau_{k+1}} y_t \sum_{t=\tau_k+1}^{\tau_{k+1}} y_{t-1}}{\Delta_k \sum_{t=\tau_k+1}^{\tau_{k+1}} y_{t-1}^2 - (\sum_{t=\tau_k+1}^{\tau_{k+1}} y_{t-1})^2} \\ \mu_k^* = \frac{\sum_{t=\tau_k+1}^{\tau_{k+1}} (y_t - \phi_k^* y_{t-1})}{\Delta_k (1 - \phi_k^*)} \end{cases}$$

3.4 Le modèle de rupture dans la tendance avec bruit AR(1)

3.4.1 Présentation du modèle

Sur le segment k la série sera modélisée par sa une droite affine de la forme $y_t = \alpha_k t + \beta_k + \varepsilon_t$ où z_t est un bruit auto-regressif $\varepsilon_t = \phi_k \varepsilon_{t-1} + z_t$ avec $z_t \sim \mathcal{N}(0, \sigma^2)$. La série s'écrit alors sous la forme :

$$y_t = \begin{cases} \alpha_0 t + \beta_0 + \varepsilon_t, & 1 \leq t \leq \tau_1 \\ \alpha_1 t + \beta_1 + \varepsilon_t, & \tau_1 + 1 \leq t \leq \tau_2 \\ \vdots \\ \alpha_K t + \beta_K, & \tau_K + 1 \leq t \leq n \end{cases}$$

3.4.2 Estimateur des paramètres du modèle

Afin de trouver les paramètres α_k , β_k et ϕ_k du modèle sur le segment $[\tau_k + 1, \tau_{k+1}]$, on va minimiser la fonction de coût obtenue comme suit. Sur le segment $[\tau_k, \tau_{k+1}]$ on a :

$$\begin{aligned} y_t &= \alpha_k t + \beta_k + \phi_k \varepsilon_{t-1} + z_t \\ &= \alpha_k t + \beta_k + \phi_k (y_{t-1} - \alpha_k (t-1) - \beta_k) + z_t \\ &= \phi_k y_{t-1} + \alpha_k (1 - \phi_k) t + \alpha_k \phi_k + \beta_k (1 - \phi_k) \end{aligned}$$

En posant $\omega_k := -\alpha_k (1 - \phi_k)$ et $\delta_k := -(\alpha_k \phi_k + \beta_k (1 - \phi_k))$ on a que $y_t - \phi_k y_{t-1} + \omega_k t + \delta_k \sim \mathcal{N}(0, \sigma^2)$. Ainsi, la fonction de coût qu'on optimisera par rapport à ϕ , δ et ω est :

$$\mathcal{C}(y, (\phi_k, \delta_k, \omega_k)) = \sum_{t=\tau_k+1}^{\tau_{k+1}} (y_t - \phi_k y_{t-1} + \omega_k t + \delta_k)^2$$

L'application $(\phi_k, \delta_k, \omega_k) \mapsto \mathcal{C}(y, (\phi_k, \delta_k, \omega_k))$ est de classe $\mathcal{C}^1(\mathbb{R}^3, \mathbb{R})$ et strictement convexe. Comme précédemment, annuler ses dérivées partielles nous donnera les uniques estimateurs ω_k^* , δ_k^* et ϕ_k^* . D'abord on a :

$$\frac{\partial \mathcal{C}(y, (\phi_k, \delta_k, \omega_k))}{\partial \phi_k} = \sum_{t=\tau_k+1}^{\tau_{k+1}} -2 y_{t-1} (y_t - \phi_k + \omega_k t + \delta_k)$$

Qui s'annule lorsque :

$$\phi_k = \frac{\sum_{t=\tau_k+1}^{\tau_{k+1}} y_{t-1}(y_t + \omega_k t + \delta_k)}{\sum_{t=\tau_k+1}^{\tau_{k+1}} y_{t-1}^2} \quad (6)$$

Ensuite :

$$\frac{\partial \mathcal{C}(y, (\phi_k, \delta_k, \omega_k))}{\partial \delta_k} = \sum_{t=\tau_k+1}^{\tau_{k+1}} 2(y_t - \phi_k y_{t-1} + \omega_k t + \delta_k)$$

Qui s'annule lorsque :

$$\delta_k = -\frac{\sum_{t=\tau_k+1}^{\tau_{k+1}} (y_t - \phi_k y_{t-1} + \omega_k t)}{\Delta_k} \quad (7)$$

Et enfin :

$$\frac{\partial \mathcal{C}(y, (\phi_k, \delta_k, \omega_k))}{\partial \omega_k} = \sum_{t=\tau_k+1}^{\tau_{k+1}} 2t(y_t - \phi_k + \omega_k t + \delta_k)$$

Qui s'annule lorsque :

$$\omega_k = -\frac{\sum_{t=\tau_k+1}^{\tau_{k+1}} t(y_t - \phi_k y_{t-1} + \delta_k)}{\sum_{t=\tau_k+1}^{\tau_{k+1}} t^2} \quad (8)$$

En injectant (8) dans (6) on obtient :

$$\phi_k = \frac{\sum_{t=\tau_k+1}^{\tau_{k+1}} t^2 \sum_{t=\tau_k+1}^{\tau_{k+1}} y_{t-1}(y_t + \delta_k) - \sum_{t=\tau_k+1}^{\tau_{k+1}} t(y_t + \delta_k) \sum_{t=\tau_k+1}^{\tau_{k+1}} t y_{t-1}}{\sum_{t=\tau_k+1}^{\tau_{k+1}} t^2 \sum_{t=\tau_k+1}^{\tau_{k+1}} y_{t-1}^2 - (\sum_{t=\tau_k+1}^{\tau_{k+1}} t y_{t-1})^2} \quad (9)$$

Et en injectant (8) dans (7) on obtient :

$$\delta_k = \frac{\sum_{t=\tau_k+1}^{\tau_{k+1}} t^2 \sum_{t=\tau_k+1}^{\tau_{k+1}} y_{t-1}(y_t - \phi_k y_{t-1}) - \sum_{t=\tau_k+1}^{\tau_{k+1}} t(y_t + \phi_k y_{t-1}) \sum_{t=\tau_k+1}^{\tau_{k+1}} t}{(\sum_{t=\tau_k+1}^{\tau_{k+1}} t)^2 - \Delta_k \sum_{t=\tau_k+1}^{\tau_{k+1}} t^2} \quad (10)$$

Finalement, en injectant (10) dans (9), on obtient l'estimateur ϕ_k^* de ϕ_k :

$$\phi_k^* = \frac{H_k J_k + I_k G_k}{F_k I_k - H_k K_k}$$

Où :

$$\left\{ \begin{array}{l} F_k := \left(\sum_{t=\tau_k+1}^{\tau_{k+1}} t^2 \right) \left(\sum_{t=\tau_k+1}^{\tau_{k+1}} y_{t-1}^2 \right) - \left(\sum_{t=\tau_k+1}^{\tau_{k+1}} t y_{t-1} \right)^2 \\ G_k := \left(\sum_{t=\tau_k+1}^{\tau_{k+1}} t^2 \right) \left(\sum_{t=\tau_k+1}^{\tau_{k+1}} y_t y_{t-1} \right) - \left(\sum_{t=\tau_k+1}^{\tau_{k+1}} t y_t \right) \left(\sum_{t=\tau_k+1}^{\tau_{k+1}} t y_{t-1} \right) \\ H_k := \left(\sum_{t=\tau_k+1}^{\tau_{k+1}} t^2 \right) \left(\sum_{t=\tau_k+1}^{\tau_{k+1}} y_{t-1} \right) - \left(\sum_{t=\tau_k+1}^{\tau_{k+1}} t \right) \left(\sum_{t=\tau_k+1}^{\tau_{k+1}} t y_{t-1} \right) \\ I_k := \left(\sum_{t=\tau_k+1}^{\tau_{k+1}} t \right)^2 - \Delta_k \sum_{t=\tau_k+1}^{\tau_{k+1}} t^2 \\ J_k := \left(\sum_{t=\tau_k+1}^{\tau_{k+1}} t^2 \right) \left(\sum_{t=\tau_k+1}^{\tau_{k+1}} y_t \right) - \left(\sum_{t=\tau_k+1}^{\tau_{k+1}} t \right) \left(\sum_{t=\tau_k+1}^{\tau_{k+1}} t y_t \right) \\ K_k := \left(\sum_{t=\tau_k+1}^{\tau_{k+1}} t \right) \left(\sum_{t=\tau_k+1}^{\tau_{k+1}} t y_{t-1} \right) - \left(\sum_{t=\tau_k+1}^{\tau_{k+1}} t^2 \right) \left(\sum_{t=\tau_k+1}^{\tau_{k+1}} y_{t-1} \right) \end{array} \right.$$

Ensuite, on obtient α_k^* l'estimateur de α_k comme $\alpha_k^* = \frac{\omega_k^*}{1-\phi_k^*}$ et β_k^* comme $\beta_k^* = -\frac{\delta^* + \alpha_k^* \phi_k^*}{1-\phi_k^*}$.

4 Détection de points de rupture

Nous nous attarderons uniquement sur des méthodes de détection de points de rupture exactes, basées sur le calcul de la fonction de coût associée à chaque segment $y_{s:t}$ de notre série temporelle. Ces algorithmes seront applicables aux cas où le nombre de points de rupture est, ou non, connu à l'avance.

4.1 Algorithme de programmation dynamique

4.1.1 Principe

L'algorithme de programmation dynamique (Optimal Segmentation en anglais) est une méthode de détection de points de rupture construite, comme son nom l'indique, autour de la notion de programmation dynamique. Définissons d'abord ce qu'est la programmation dynamique.

La programmation dynamique a pour principe de résoudre un problème en le décomposant en plusieurs sous-problèmes plus facilement traitables. Il s'agit alors de traiter ces sous-problèmes du plus petit au plus important en stockant les résultats de chacun d'entre eux.

L'algorithme de programmation dynamique est adapté à la fois aux problèmes où le nombre de points de rupture (noté K par la suite) est connu ou inconnu. Les 2 cas se traitant quasiment de la même manière, nous vous expliquons d'abord le principe de cet algorithme.

L'idée est de minimiser la fonction de coût sachant K . La solution optimale peut être calculée grâce à l'additivité de la fonction de coût. En effet, en posant :

- $\mathcal{C}_{1:n}^K$ le coût optimal de la décomposition de $y_{1:n}$ en $K + 1$ segments
- $\mathcal{T}_n = \{\tau_{0:K+1} : \tau_0 < \tau_1 < \dots < \tau_K < \tau_{K+1} = n\}$, l'ensemble des configurations possibles de K points de rupture de la série $y_{1:n}$.

Nous remarquons :

$$\begin{aligned}\mathcal{C}_{1:n}^K &= \min_{\tau \in \mathcal{T}_n} \sum_{k=1}^{K+1} \mathcal{C}(y_{\tau_{k-1}+1:\tau_k}) \\ &= \min_{\tau \in \mathcal{T}_n} \left(\sum_{k=1}^K [\mathcal{C}(y_{\tau_{k-1}+1:\tau_k})] + \mathcal{C}(y_{\tau_K+1:n}) \right) \\ &= \min_{t \leq n - \tau_K} (\mathcal{C}_{1:t}^{K-1} + \mathcal{C}(y_{t+1:n}))\end{aligned}$$

Notre problème se transforme alors en une multitude de problèmes de minimisation de fonction de coût à une inconnue (le dernier point de rupture) conditionnellement au fait que nous connaissions l'ensemble des précédents points de rupture ("précédent" ici est au sens temporel). Pour déterminer l'ensemble des points de rupture de notre série temporelle, il suffit d'appliquer cette stratégie récursivement par rapport aux points de rupture.

4.1.2 Implémentation sous R

Comme énoncé dans la partie ci-dessus, le calcul d'un $k^{\text{ème}}$ point de rupture s'avère facile si la segmentation des $(k - 1)^{\text{ème}}$ précédents points de rupture sur chaque sous segment $y_{1:t}$ de la série est connu. C'est le principe d'additivité. L'idée ici est donc à chaque étape de calculer l'ensemble des coûts associés à l'ajout

d'un $k^{\text{ème}}$ point de rupture sur chaque segment de la CET de la forme $y_{1:t}$ afin de pouvoir déterminer le $(k+1)^{\text{ème}}$ point de rupture. Nous retombons alors sur la notion de programmation dynamique.

L'implémentation de l'algorithme de programmation dynamique se fait par récurrence. L'initialisation se traduit par le calcul de la fonction de coût sur chaque segment de la série temporelle.

Introduisons $\tau_{k,i}$ le $i^{\text{ème}}$ point de rupture de la CET sachant qu'il y a k points de rupture, $k \leq K, i \leq k$.

A chaque étape de la récurrence, l'algorithme calcule le coût qui minimise l'ajout d'un $k^{\text{ème}}$ point de rupture sur tous les segments de la forme $y_{1:j}$ sachant que $k-1$ sont déjà placés sur un sous segment $y_{1:v}$, $v < j$. Il va ensuite déterminer le point qui minimise le coût induit par l'ajout d'un $k^{\text{ème}}$ point de rupture sur $y_{1:n}$, sachant que $k-1$ sont déjà placés sur un sous segment $y_{1:v}$, $v < n$. Ce point sera le $k^{\text{ème}}$ point de rupture.

La dernière étape est de calculer les $k-1$ points de rupture restant comme suit : Pour le $t^{\text{ème}}$ point de rupture, il s'agit du point minimisant l'ajout d'un point de rupture sur $y_{1:\tau_{k,t+1}}$ sachant qu'il y a $t-1$ points de rupture sur un sous segment $y_{1:v}$ de $y_{1:\tau_{t+1}}$.

Nous obtenons alors le pseudo-code ci-après :

Algorithme 1 : Algorithme de Programmation Dynamique

Entrées : y une liste de taille n , \mathcal{C} une fonction de cout, K_{max} le nombre de points de rupture maximal désiré.

Sorties : Listes des k points de rupture $\tau_{k,1:k}$ et le coût optimal $q_{1:n}^k$ associé, $k \in \{1, \dots, K_{max}\}$.

```

1 initialisation :  $q_{i,j}^0 = \mathcal{C}(y_{i:j}), \forall i, j \in \{1, \dots, n\}, i \leq j$  la matrice des coûts de chaque segment de  $y$ .
2 pour  $k = 1, \dots, K_{max}$  faire
3   pour  $j = 1, \dots, n$  faire
4      $q_{1:j}^k = \min_{v \in \{1, \dots, j-1\}} (q_{1,v}^{k-1} + q_{v+1,j}^0)$ 
5    $\tau_{k,1} = \arg \min_{v \in \{1, \dots, n-1\}} (q_{1,v}^{k-1} + q_{v+1,n}^0)$ 
6   pour  $i = 2, \dots, k$  faire
7      $\tau_{k,i} = \arg \min_{v \in \{1, \dots, n\}} (q_{1,v}^{k-i} + q_{v+1,\tau_{k,i-1}}^0)$ 

```

Dans un cas où nous connaissons le nombre de points de rupture à l'avance (supposons que nous en cherchons $S, S \leq K_{max}$), nous ne gardons des sorties de l'algorithme de programmation dynamique que la liste $\tau_{S,1:S}$, correspondant aux S points de ruptures optimaux, ainsi que le coût optimal $q_{1:n}^S$ associé à la segmentation de la CET par $\tau_{S,1:S}$.

Si le nombre de points de rupture souhaité n'est pas fixé par l'utilisateur, il est alors nécessaire de pénaliser les modèles. Après avoir fait tourner l'algorithme de programmation dynamique, il suffit d'appliquer une pénalité sur l'ensemble des $(q_{1:n}^K)_{K=1, \dots, K_{max}}$ renvoyés par le programme, puis de sélectionner la liste des points de ruptures dont le coût pénalisé est optimal.

4.1.3 Complexité

L'inconvénient majeur de l'algorithme de programmation dynamique est sa complexité quadratique en $\mathcal{O}(n^2)$. Ce problème était attendu puisque l'initialisation en elle même nécessite de parcourir les $\frac{n^2}{2}$ segments de notre série temporelle.

En somme, cet algorithme contient :

- 2 boucles imbriquées en $\mathcal{O}(n)$ à l'initialisation ;
- Une boucle en $\mathcal{O}(K)$ contenant une boucle ayant une complexité en $\mathcal{O}(n^2)$ (du fait du *min* en $\mathcal{O}(n)$), un calcul en $\mathcal{O}(n)$ puis une instruction en $\mathcal{O}(n)$.

D'où la complexité finale en $\mathcal{O}(n^2)$.

4.1.4 Applications numériques et Résultats

Nous avons appliqué l'algorithme de programmation dynamique sur la CET avec nos 4 différents modèles de la partie 3. Nous ne vous partageons ici que les résultats des modèles pénalisés (le cas où le nombre de points de rupture est connu n'étant pas très intéressant à observer) avec $K_{max} = 10$. Pour le modèle de rupture dans la moyenne, nous obtenons 3 points de rupture : en 1701, 1919 et en 1988. Nous vous expliquons ci-après le raisonnement nous permettant d'arriver à ces points de rupture :

Tout d'abord, voici la matrice des listes de points de rupture pour un nombre de points de rupture allant de 1 à 10.

```
> Liste_rupt_mod1
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
[1,]  329    0    0    0    0    0    0    0    0    0
[2,]  329   43    0    0    0    0    0    0    0    0
[3,]  330  261   43    0    0    0    0    0    0    0
[4,]  330  261   40   32    0    0    0    0    0    0
[5,]  330  252   82   81   47    0    0    0    0    0
[6,]  330  252   82   81   43   29    0    0    0    0
[7,]  330  252   82   81   68   40   32    0    0    0
[8,]  330  234  226   82   81   68   40   32    0    0
[9,]  330  234  221  220   82   81   68   40   32    0
[10,] 330  234  220  158  153   82   81   68   40   32
```

A laquelle est associée la liste des coûts optimaux :

```
> Liste_cout_opt_mod1
[1] 132.16447 119.70124 113.74801 109.06641 105.73299 101.52254 97.89367
[8] 95.27968 93.73158 91.29198
```

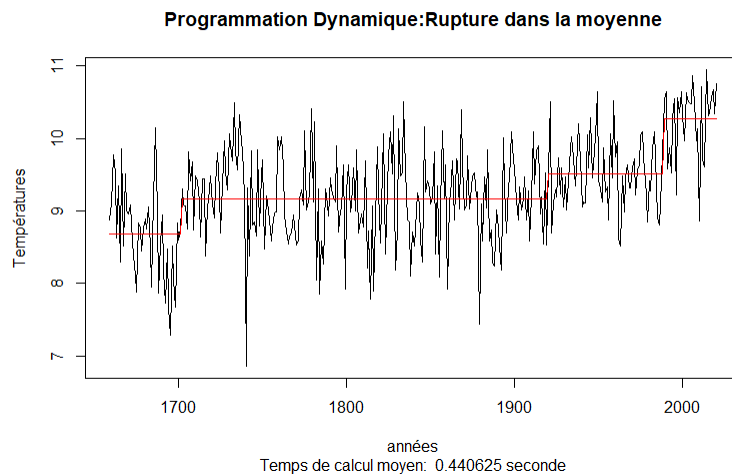
En pénalisant par le BIC chaque élément indicé par k de cette liste de coûts par $((k + 1) + \log(362))$, nous obtenons :

```
> Liste_cout_opt_pen_mod1
[1] 143.9478 137.3762 137.3146 138.5246 141.0829 142.7640 145.0268 148.3045
[9] 152.6480 156.1001
```

Le minimum est atteint pour 3 points de rupture qui sont finalement :

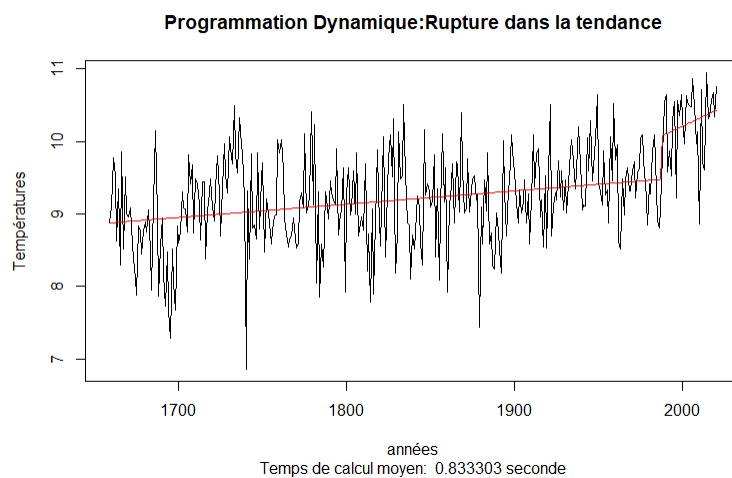
```
> rupture_mod_mean
[1] 43 261 330
> annee[rupture_mod_mean]
[1] 1701 1919 1988
```

Avec le graphe associé :



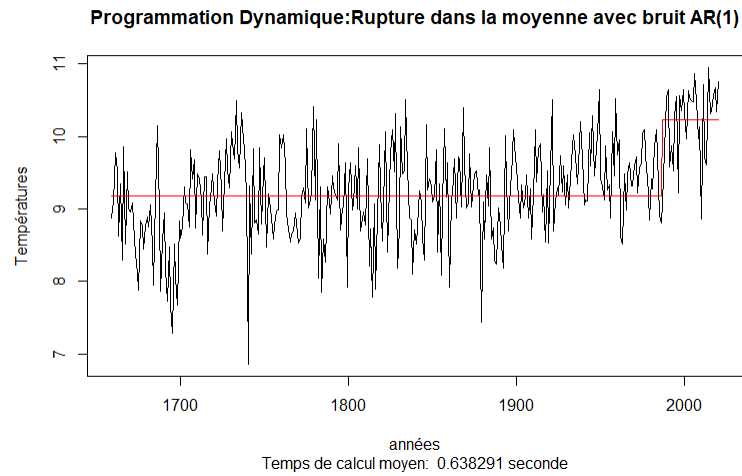
Pour le modèle de rupture dans la tendance, nous n'obtenons qu'un unique point de rupture en 1987 :

```
> rupture_mod_trend
[1] 329
> annee[rupture_mod_trend]
[1] 1987
```



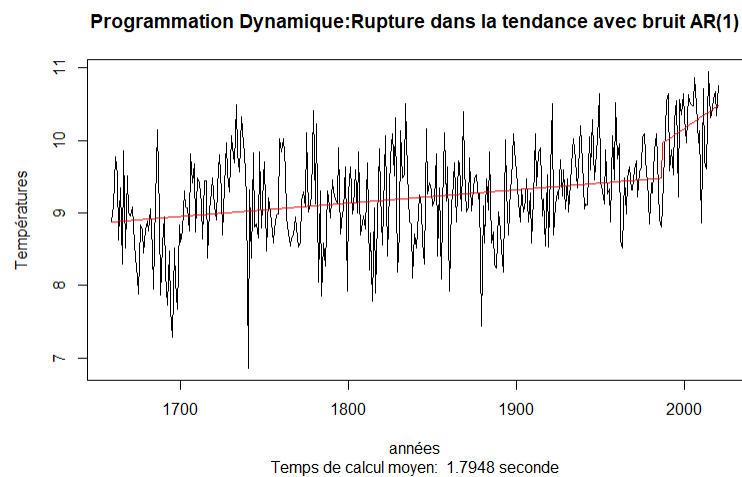
Pour le modèle de rupture dans la moyenne avec un bruit AR(1), nous obtenons aussi un point de rupture en 1986 :

```
> rutpure_mean_mod3
[1] 328
> annee[rutpure_mean_mod3]
[1] 1986
```



Pour le modèle de rupture dans la tendance avec un bruit AR(1), nous obtenons le même point de rupture que pour le 3^{ème} modèle :

```
> rutpure_mean_mod3
[1] 328
> annee[rutpure_mean_mod3]
[1] 1986
```



4.2 Algorithme PELT

4.2.1 Principe

L'algorithme PELT (Pruned Exact Linear Time) est un algorithme de détection de points de rupture adapté aux problèmes où le nombre de point de rupture est inconnu. Comme l'algorithme de programmation dynamique, l'objectif est de minimiser le coût et la pénalité liée au fait d'avoir k points de rupture :

$$\underbrace{\sum_{i=1}^{k+1} \left[\mathcal{C}(y_{\tau_{i-1}+1: \tau_i}) \right]}_{\text{le coût}} + \underbrace{\text{pen}(k)}_{\text{la pénalité}}$$

L'algorithme PELT a été conçu pour une fonction de pénalité linéaire. C'est-à-dire les fonctions de la forme :

$$\mathcal{P} : \begin{cases} \mathbb{N} & \longrightarrow \mathbb{R} \\ k & \longmapsto \beta(k+1) \end{cases}$$

où $\beta \in \mathbb{R}^+$ On obtient donc :

$$\sum_{i=1}^{k+1} \left[\mathcal{C}(y_{\tau_{i-1}+1: \tau_i}) \right] + \beta(k+1)$$

En introduisant le terme de pénalité dans la somme, nous obtenons :

$$\sum_{i=1}^{k+1} \left[\mathcal{C}(y_{\tau_{i-1}+1: \tau_i}) + \beta \right]$$

Pour comprendre le principe de l'algorithme PELT, nous allons introduire :

- K_{max} le nombre maximal de points de rupture.
- $\mathcal{T}_t = \{\tau_{0:K+1} : \tau_0 < \tau_1 < \dots < \tau_K < \tau_{K+1} = t, K = 1, \dots, K_{max}\}$, l'ensemble des configurations possibles de K points de rupture pour $K = 1, \dots, K_{max}$ de la série $y_{1:t}$.
- $F(s) = \min_{\tau \in \mathcal{T}_s} \sum_{k=1}^{K_s^{opt}+1} [\mathcal{C}(y_{\tau_{k-1}+1: \tau_k}) + \beta]$. On note K_s^{opt} le nombre de points de rupture qui permet d'atteindre ce minimum.

Soit $t \in [0, s]$:

$$F(s) = \min_{\tau \in \mathcal{T}_s} \sum_{k=1}^{K_s^{opt}+1} [\mathcal{C}(y_{\tau_{k-1}+1: \tau_k}) + \beta] \quad (11)$$

$$= \min_t \left\{ \min_{\tau \in \mathcal{T}_t} \sum_{k=1}^{K_t^{opt}} [\mathcal{C}(y_{\tau_{k-1}+1: \tau_k}) + \beta] + \mathcal{C}(y_{(t+1):s}) + \beta \right\} \quad (12)$$

$$= \min_t \{ F(t) + \mathcal{C}(y_{(t+1):s}) + \beta \} \quad (13)$$

Donc s'il existe un t tel que :

$$F(t) + \mathcal{C}(y_{(t+1):s}) + \beta > F(s)$$

Alors t ne peut pas être le dernier point de rupture optimal de la série $y_{1:s}$. Grâce à ce constat, on peut donc éliminer les points de rupture non optimaux et ne garder que les points de ruptures qui peuvent potentiellement être optimaux, nous allons les appeler points de rupture admissibles. C'est le principe d'élagage de l'algorithme PELT.

4.2.2 Implémentation sous R

Pour l'implémentation de l'algorithme nous allons traduire le principe d'élagage par la récurrence suivante :

Nous cherchons les points de rupture optimaux de la série $y_{1:n}$. Nous introduisons R_s , pour $s = 1, \dots, n$, l'ensemble des points de rupture admissibles pour être le dernier point de rupture de la série $y_{1:s}$. Par le principe d'élagage on définit :

$$R_{s+1} = \left\{ t \in R_s, F(t) + \mathcal{C}(y_{(t+1):s}) + \beta \leq F(s) \right\} \cup \{s\}$$

Nous remarquons que :

- R_{s+1} n'est jamais vide, il contient au moins l'élément s .
- le nombre optimal de points de rupture est déterminé par l'algorithme. En effet pour chaque s on pose :
 - cp est une liste de la même longueur que la série. Pour chaque élément s de la série nous allons stocker dans cette liste le dernier point de rupture optimal de la sous série $y_{1:s}$
 - Nous l'initialisons par $cp[1] = 0$
 - $cp[s] = \tilde{s}$, où $\tilde{s} = \arg \min_{t \in R_s} \{F(t) + \mathcal{C}(y_{(t+1):s}) + \beta\}$

Lors de l'implémentation de l'algorithme, nous nous sommes rendu compte que pour la CET la condition pour obtenir les points de rupture admissibles était trop restrictive, nous l'avons assouplie en posant :

$$R_{s+1} = \left\{ t \in R_s, F(t) + \mathcal{C}(y_{(t+1):s}) \leq F(s) \right\} \cup \{s\}$$

L'algorithme PELT parcourt la série et cherche pour chaque élément le dernier point de rupture optimal. Nous pouvons remarquer que pour appliquer cet algorithme il est nécessaire de définir une distance minimale entre les points de rupture consécutifs. En effet, nous ne pouvons pas débiter l'algorithme sur le premier élément de la série, nous devons débiter l'algorithme à $2 \times$ la distance minimale puis commencer à chercher le dernier point de rupture optimal à partir du premier élément à la distance minimale du premier élément de la série. En procédant ainsi, nous devons modifier l'initialisation de notre algorithme. Nous posons :

- $long.min$ la distance minimale entre deux points de rupture consécutifs
- Pour i allant de 1 à $2 \times long.min$ $cp[i] = 0$
- $R_1 = \{long.min\}$

Nous pouvons récupérer les points de rupture optimaux de la série complète $y_{1:n}$ grâce à la liste cp . En effet nous avons stocké le dernier point de rupture de la série dans $cp[n]$, pour récupérer le point de rupture précédent, il suffit de remonter au $cp[n]$ ième élément de la série et regarder le dernier point de rupture détecté par le point précédent et ainsi de suite jusqu'à arriver au premier élément de la série. Nous obtenons le pseudo-code suivant :

Algorithme 2 : Algorithme PELT

Entrées : y une liste de taille n , \mathcal{C} une fonction de coût, β la pénalité, $long.min$ la longueur minimale entre les points de rupture.

Sorties : $optcp$ Liste contenant les points de rupture optimaux

```
1 initialisation :  $R_1 = \{long.min\}$ ,  $cp[1 : 2 \times long.min] \leftarrow 0$ 
2 pour  $s = 1, \dots, n$  faire
3    $F(s) = \min_{t \in R_s} \{F(t) + \mathcal{C}(y_{(t+1):s}) + \beta\}$ ;
4    $\tilde{s} = \arg \min_{t \in R_s} \{F(t) + \mathcal{C}(y_{(t+1):s}) + \beta\}$  ;
5    $cp[s] \leftarrow \tilde{s}$  ;
6    $R_{s+1} = \left\{ t \in R_s, F(t) + \mathcal{C}(y_{(t+1):s}) \leq F(s) \right\} \cup \{s\}$  ;
7  $i \leftarrow L$  ;
8  $optcp \leftarrow c()$ ;
9 tant que  $i > 1$  faire
10    $optcp \leftarrow c(optcp, cp[i])$  ;
11    $i \leftarrow cp[i] - 1$ 
```

4.2.3 Complexité

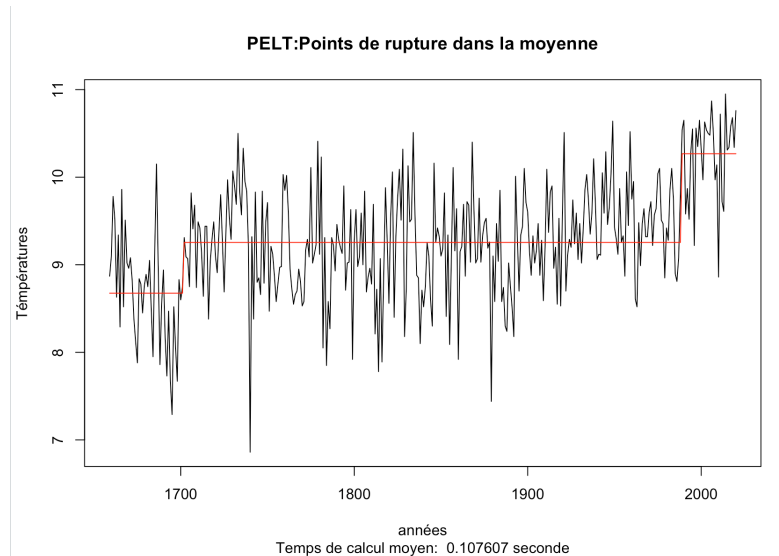
L'avantage de l'algorithme PELT comparé à l'algorithme de programmation dynamique est sa complexité. En effet, l'algorithme PELT a une complexité linéaire comparée à la complexité en $\mathcal{O}(n^2)$ de l'algorithme de programmation dynamique. En analysant le code de l'algorithme PELT, nous remarquons qu'il n'y a qu'une seule boucle de complexité $\mathcal{O}(n)$. Pour avoir une complexité linéaire en la longueur de la série étudiée, il faut que la recherche du min sur R_s soit en $\mathcal{O}(1)$. Nous pouvons supposer que cette hypothèse est vérifiée. En effet, grâce au principe d'élagage, R_s reste de petite taille. Nous pouvons le constater expérimentalement en faisant tourner l'algorithme sur la CET. D'où la complexité linéaire de l'algorithme. Ces intuitions sur les conditions nécessaires pour avoir une complexité linéaire sont formalisées mathématiquement dans le théorème 3.2 de l'article de R. Killick, P. Fearnhead et I.A. Eckley.[2]

4.2.4 Applications numériques et Résultats

Nous avons testé l’algorithme PELT sur la CET.

Pour le modèle de rupture dans la moyenne, nous obtenons 2 points de rupture, en 1701 et en 1988 :

```
> rupture_mod_mean
[1] 43 330
> annee[rupture_mod_mean]
[1] 1701 1988
```



Nous pouvons retrouver les points de rupture détectés par l'algorithme PELT en observant la liste *optcp* de l'algorithme.

[illegible]

FIGURE 8 – Matrice des derniers points de rupture optimaux pour tous les éléments de la série CET

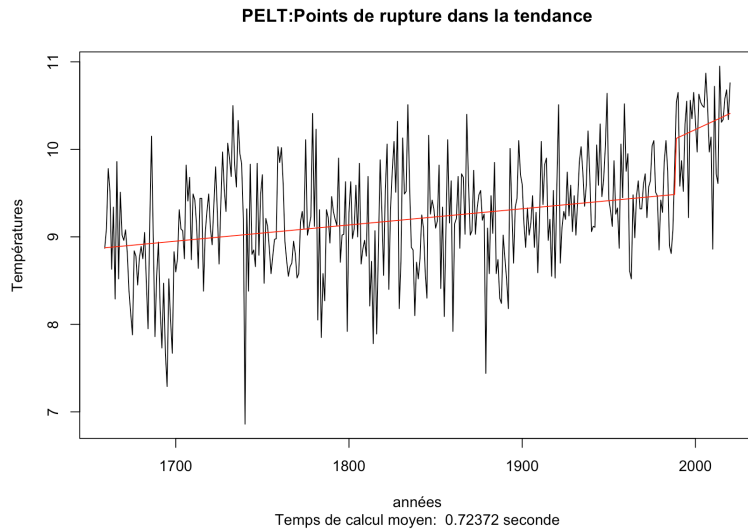
Nous partons du dernier élément de la liste, il s'agit de 330 qui est encadré en rouge sur la figure 8, il s'agit d'un point de rupture. Nous remontons jusqu'au 330 ième élément, qui correspond à 252 (encadré en rouge) et nous regardons le dernier point de rupture de l'élément précédent (souligné en rouge), il s'agit de 43 (encadré en rouge). C'est un point de rupture. Enfin nous remontons au 43 ième élément de la liste, son dernier point de rupture est 1. Nous en concluons que l'algorithme PELT a détecté 2 points de rupture : 43 et 330.

Pour le modèle de rupture dans la tendance, nous obtenons 1 seul point de rupture, en 1988 :


```

> rupture_mod_trend
[1] 330
> annee[rupture_mod_trend]
[1] 1988

```

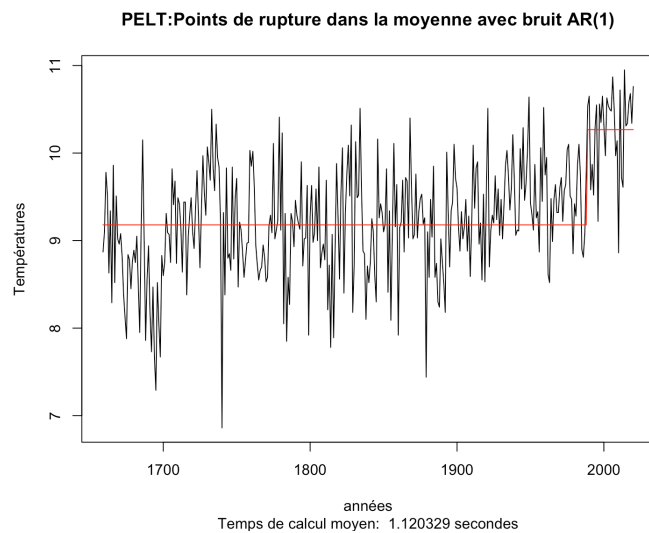


Pour le modèle de rupture dans la moyenne avec un bruit AR(1), nous obtenons également 1 seul point de rupture, en 1988 :

```

> rupture_mean_mod3
[1] 330
> annee[rupture_mean_mod3]
[1] 1988

```



Pour le modèle de rupture dans la tendance avec un bruit AR(1), l'algorithme PELT n'a détecté aucun point de rupture.

4.3 Comparaison de l'algorithme de programmation dynamique et de l'algorithme PELT

Dans cette partie, nous comparons l'algorithme PELT et l'algorithme de programmation dynamique sur des simulations.

4.3.1 Complexité

Dans un premier temps nous avons comparé la complexité des deux algorithmes à l'aide d'un graphe sur lequel nous avons représenté en abscisse la taille de la série simulée, et en ordonnée le temps d'exécution des algorithmes.

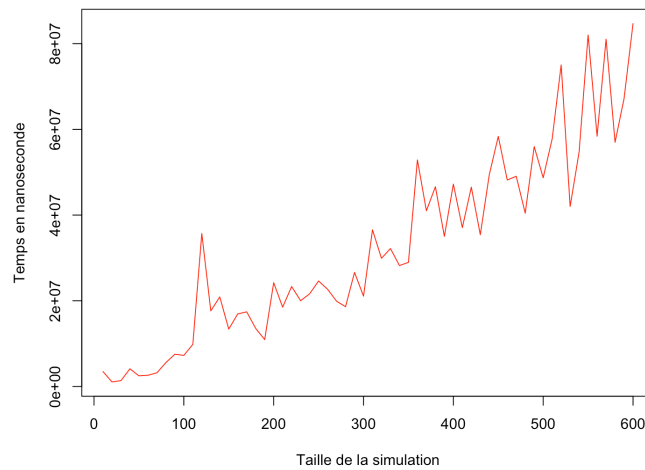


FIGURE 9 – Temps d'exécution de l'algorithme PELT en fonction de la taille de la série (série simulée)

Nous voyons se dessiner une droite qui illustre la complexité en $\mathcal{O}(n)$ de l'algorithme PELT.

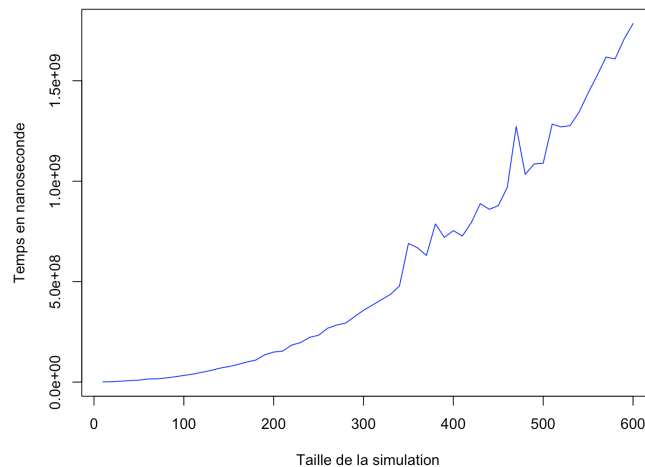


FIGURE 10 – Temps d'exécution de l'algorithme de programmation dynamique en fonction de la taille de la série (série simulée)

Nous voyons se dessiner une parabole qui illustre la complexité en $\mathcal{O}(n^2)$ de l'algorithme de programmation dynamique.

Nous pouvons superposer les deux graphes pour comparer les temps d'exécution des deux algorithmes.

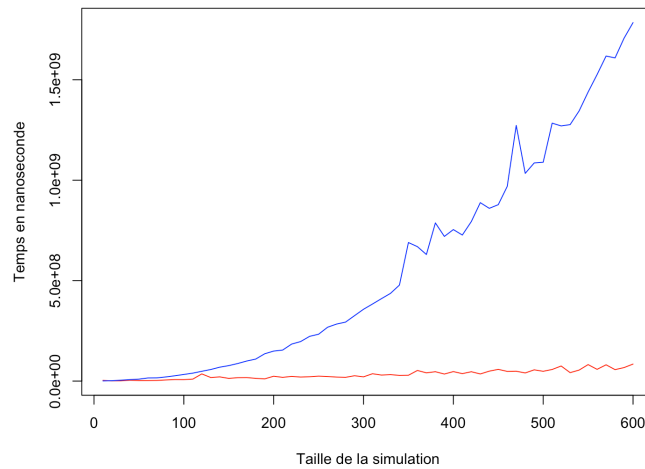


FIGURE 11 – Temps d'exécution des deux algorithmes en fonction de la taille de la série (série simulée). L'algorithme PELT est représenté en rouge et l'algorithme de programmation dynamique est représenté en bleu

Nous voyons très clairement que l'algorithme PELT est beaucoup plus rapide que l'algorithme de programmation dynamique.

4.3.2 Simulations

Dans cette partie nous allons présenter les résultats de l'algorithme de programmation dynamique et de l'algorithme PELT sur des données simulées suivant nos 4 modèles.

Modèle de rupture dans la moyenne

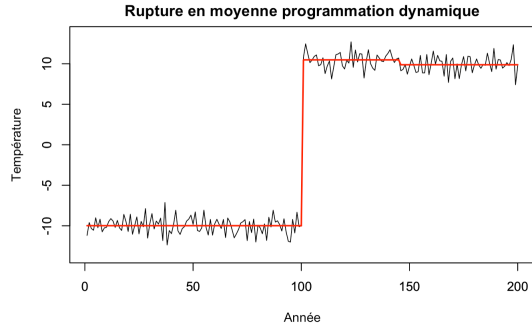


FIGURE 12 – Représentation graphique des résultats de la programmation dynamique sur des simulations

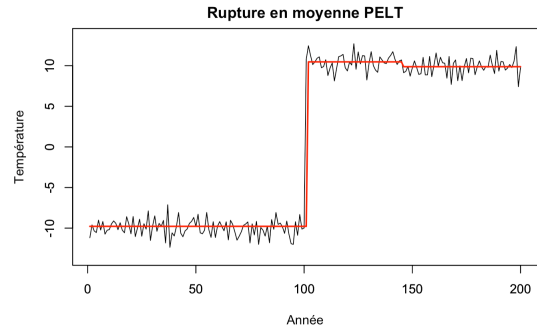


FIGURE 13 – Représentation graphique des résultats du PELT sur des simulations

Modèle de rupture dans la tendance

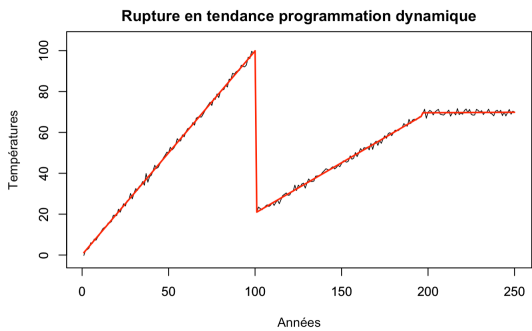


FIGURE 14 – Représentation graphique des résultats de la programmation dynamique sur des simulations

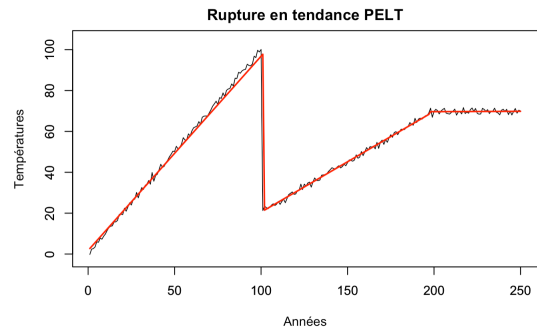


FIGURE 15 – Représentation graphique des résultats du PELT sur des simulations

Modèle de rupture dans la moyenne avec AR(1)

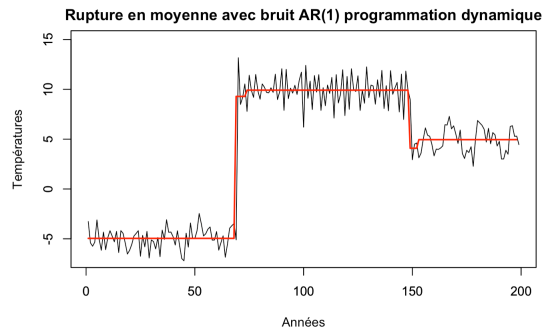


FIGURE 16 – Représentation graphique des résultats de la programmation dynamique sur des simulations

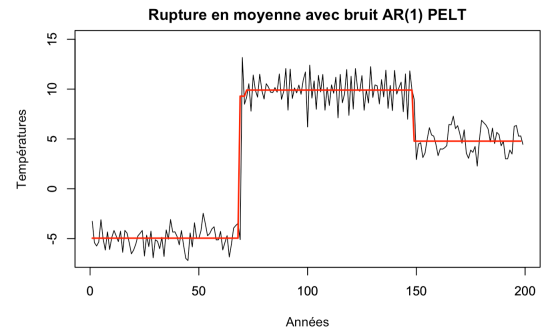


FIGURE 17 – Représentation graphique des résultats du PELT sur des simulations

Modèle de rupture dans la tendance avec AR(1)

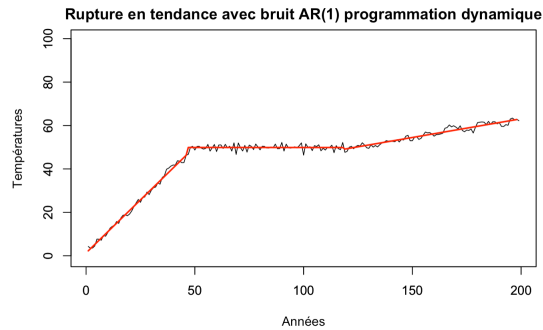


FIGURE 18 – Représentation graphique des résultats de la programmation dynamique sur des simulations

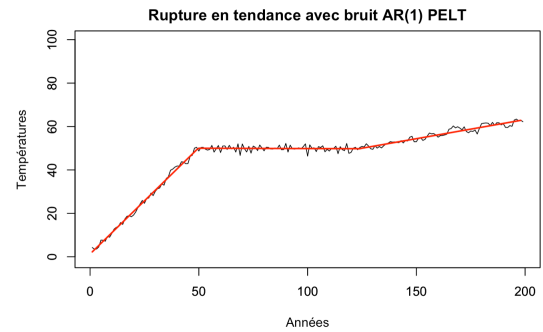


FIGURE 19 – Représentation graphique des résultats du PELT sur des simulations

Conclusion

Nous avons pour objectif de mettre en place des modèles descriptifs nous permettant d'étudier les propriétés de la série CET. Grâce aux quatre modèles que nous avons élaborés, nous constatons une dynamique de hausse des températures. En effet, dans les modèles de rupture dans la moyenne, nous observons des paliers ascendants et dans les modèles de rupture dans la tendance nous observons à la fois des paliers ascendants et sur chaque palier, une tendance à la hausse.

En pénalisant nos algorithmes, nous avons pu ne considérer que les points de ruptures les plus significatifs. Ainsi les plus gros changements de température s'opèrent en 1988 et dans une moindre mesure en 1701.

Du point de vue algorithmique, nous avons implémenté l'algorithme de programmation dynamique et l'algorithme PELT pour la détection de point de rupture. L'algorithme PELT a pour avantage sa complexité linéaire, sa rapidité d'exécution est un véritable atout. Cependant, l'algorithme PELT intègre une pénalité linéaire en le nombre de paramètre à estimer. Lorsque le modèle se complexifie et que le nombre de paramètre à estimer devient important, la pénalité a tendance à augmenter. De plus avec le principe d'élagage de l'algorithme PELT et la distance minimale imposée entre les points de rupture consécutifs, lorsque nous considérons des modèles très complexes, aucun point de rupture n'est détecté. C'est le cas avec le modèle 4, le modèle de rupture dans la tendance avec un bruit $AR(1)$.

Bibliographie

- [1] G.J. JENKINS, M.C. PERRY et M.J. PRIOR. *The climate of the United Kingdom and recent trends*. UK : Met Office Hadley Centre Exeter, 2008.
- [2] Rebecca KILLICK et Idris.A ECKLEY. “Changepoint : An R Package for Changepoint Analysis”. In : *Journal of statistical software* 58 (2014), Issue 3.
- [3] Emile LEBARBIER et Mary-Huard TRISTAN. “Une introduction au critère BIC :fondements théoriques et interprétation”. In : *Journal de la société française de statistique* 157 (2006), p. 39-57.
- [4] Xueheng SHI, Claudie BEAULIEU et Killick REBECCA. *Changepoint Detection : An Analysis of the Central England Temperature Series*. Juin 2021.
- [5] Charles TRUONG, Laurent OUDRE et Nicolas VAYATIS. *Optimal detection of changepoints with a linear computational cost*. [http:https://arxiv.org/pdf/1101.1438.pdf?ref=https://githubhelp.com](https://arxiv.org/pdf/1101.1438.pdf?ref=https://githubhelp.com). Oct. 2012.
- [6] Charles TRUONG, Laurent OUDRE et Nicolas VAYATIS. *Selective review of offline point detection methods*. <http://www.elsevier.com/locate/sigpro>. Sept. 2019.