

Séries Temporelles

Notes de Cours

Master M1 — 2025–2026

Yaë Ulrich Gaba

« La prédiction est un art difficile, surtout lorsqu'elle concerne l'avenir. »

— Niels Bohr

Table des matières

Préface	vii
1 Introduction aux Séries Temporelles	1
1.1 Exemple motivant	1
1.2 Définitions fondamentales	1
1.3 Composantes d'une série temporelle	1
1.4 Opérateur retard et différence	2
1.5 Exemples de séries temporelles	2
1.6 Visualisation	2
1.7 Implémentation	3
1.8 Exercices	3
2 Stationnarité et Autocovariance	5
2.1 Exemple motivant	5
2.2 Stationnarité stricte et faible	5
2.3 Fonction d'autocovariance	5
2.4 Estimation de l'autocovariance	6
2.5 Autocorrélation partielle	6
2.6 Tests de stationnarité	6
2.7 Implémentation	7
2.8 Exercices	7
3 Modèles AR, MA, ARMA	9
3.1 Exemple motivant	9
3.2 Processus autorégressifs $AR(p)$	9
3.3 Équations de Yule–Walker	10
3.4 Processus moyenne mobile $MA(q)$	10
3.5 Processus $ARMA(p, q)$	10
3.6 Signatures ACF/PACF	11
3.7 Implémentation	11
3.8 Exercices	12
4 Identification, Estimation et Diagnostic	13
4.1 Exemple motivant	13
4.2 La méthodologie de Box–Jenkins	13
4.3 Identification	13
4.3.1 Examen de l'ACF et de la PACF	13
4.3.2 Critères d'information	14
4.4 Estimation	14

4.4.1	Maximum de vraisemblance	14
4.4.2	Moindres carrés conditionnels	14
4.5	Diagnostic	14
4.5.1	Analyse des résidus	14
4.6	Implémentation	15
4.7	Exercices	16
5	Modèles ARIMA et Différenciation	17
5.1	Exemple motivant	17
5.2	Processus intégrés	17
5.3	Modèle ARIMA(p, d, q)	17
5.4	SARIMA : saisonnalité	17
5.5	Stratégie de différenciation	18
5.6	Exemple numérique : AirPassengers	18
5.7	Implémentation	18
5.8	Exercices	19
6	Modèles GARCH et Volatilité Conditionnelle	21
6.1	Exemple motivant	21
6.2	Faits stylisés des rendements financiers	21
6.3	Modèle ARCH	21
6.4	Modèle GARCH	22
6.5	IGARCH	22
6.6	Extensions	22
6.6.1	EGARCH	22
6.6.2	GJR-GARCH	22
6.7	Estimation	23
6.8	Tests d'effet ARCH	23
6.9	Implémentation	23
6.10	Exercices	24
7	Analyse Spectrale	25
7.1	Exemple motivant	25
7.2	Densité spectrale	25
7.3	Spectres de processus classiques	25
7.4	Périodogramme	26
7.5	Estimation spectrale	26
7.5.1	Fenêtre spectrale (lissage)	26
7.5.2	Méthode paramétrique	26
7.6	Implémentation	27
7.7	Exercices	28
8	Modèles à Espace d'États et Filtre de Kalman	29
8.1	Exemple motivant	29
8.2	Modèle linéaire gaussien à espace d'états	29
8.3	Filtre de Kalman	29
8.4	Lisseur de Kalman	30
8.5	Vraisemblance par le filtre de Kalman	30
8.6	Implémentation	30

8.7	Exercices	31
9	Séries Temporelles Multivariées — VAR et Coïntégration	33
9.1	Exemple motivant	33
9.2	Processus vectoriels stationnaires	33
9.3	Modèle VAR(p)	33
9.3.1	Estimation	34
9.4	Causalité de Granger	34
9.5	Fonctions de réponse impulsionnelle	34
9.6	Coïntégration	34
9.6.1	Test de Johansen	34
9.7	Implémentation	35
9.8	Exercices	35
10	Prévision et Intervalles de Confiance	37
10.1	Exemple motivant	37
10.2	Prévision optimale	37
10.3	Prévision ARMA	37
10.4	Intervalles de prévision	38
10.5	Prévision ARIMA	38
10.6	Critères d'évaluation	38
10.7	Test de Diebold–Mariano	38
10.8	Implémentation	39
10.9	Exercices	40
11	Applications : Finance, Économie, Météorologie	41
11.1	Exemple motivant	41
11.2	Finance : Modélisation de la volatilité	41
11.2.1	Valeur à risque (VaR)	41
11.2.2	Application	41
11.3	Économie : Prévision macroéconomique	42
11.3.1	PIB et indicateurs avancés	42
11.4	Météorologie : Températures et saisonnalité	43
11.5	Résumé méthodologique	44
11.6	Exercices	44
	Formulaire	45
.1	Processus ARMA	45
.2	Autocovariance et autocorrélation	45
.3	Critères d'information	45
.4	GARCH	45
.5	Filtre de Kalman	45
.6	VAR	46

Préface

Les données indexées par le temps sont omniprésentes : cours boursiers minute par minute, températures journalières, PIB trimestriel, signaux EEG, trafic réseau. L'**analyse des séries temporelles** fournit le cadre mathématique pour modéliser la *dépendance temporelle* inhérente à ces données, et exploiter cette structure pour la *prévision*, la *détection d'anomalies* et la *compréhension* des mécanismes sous-jacents.

Ce cours, destiné aux étudiants de Master en statistique, économie et science des données, couvre la théorie classique (processus ARMA, analyse spectrale, filtre de Kalman) et les développements modernes (GARCH, cointégration, modèles VAR). Chaque concept est illustré par des données réelles et accompagné d'implémentations en Python.

Prérequis. Probabilités (convergences, espérance conditionnelle), statistique mathématique (estimation, tests), analyse réelle, algèbre linéaire.

Références.

- BOX, Jenkins, Reinsel, Ljung — *Time Series Analysis : Forecasting and Control*, Wiley.
- BROCKWELL, Davis — *Introduction to Time Series and Forecasting*, Springer.
- HAMILTON — *Time Series Analysis*, Princeton University Press.
- SHUMWAY, Stoffer — *Time Series Analysis and Its Applications*, Springer.
- GOURIÉROUX, Monfort — *Séries Temporelles et Modèles Dynamiques*, Economica.

Chapitre 1

Introduction aux Séries Temporelles

1.1 Exemple motivant

Le cours de clôture journalier de l'indice CAC 40 de 2000 à 2024 présente une tendance, des cycles et une volatilité variable. Contrairement aux données iid, chaque observation dépend des précédentes. L'analyse des séries temporelles exploite cette dépendance.

1.2 Définitions fondamentales

Définition 1.1 (Série temporelle). Une **série temporelle** (ou **processus stochastique à temps discret**) est une suite de variables aléatoires $(X_t)_{t \in \mathbb{Z}}$ définies sur un espace de probabilité $(\Omega, \mathcal{F}, P)$. Une **réalisation** (ou trajectoire) est une suite observée $(x_1, \dots, x_n)$.

Définition 1.2 (Bruit blanc). Un **bruit blanc** (ε_t) est une suite de variables aléatoires telles que :

1. $\mathbb{E}[\varepsilon_t] = 0$ pour tout t ,
2. $\text{Var}(\varepsilon_t) = \sigma^2$ pour tout t ,
3. $\text{Cov}(\varepsilon_t, \varepsilon_s) = 0$ pour $t \neq s$.

Notation : $\varepsilon_t \sim \text{BB}(0, \sigma^2)$. Si de plus les ε_t sont gaussiens, on parle de **bruit blanc gaussien**.

Définition 1.3 (Bruit blanc fort). Un **bruit blanc fort** est une suite iid de variables aléatoires de moyenne nulle et variance finie. Un bruit blanc (faible) n'impose que la décorrélation, pas l'indépendance.

1.3 Composantes d'une série temporelle

Définition 1.4 (Décomposition classique). On décompose souvent une série en :

$$X_t = T_t + S_t + R_t,$$

où T_t est la **tendance** (trend), S_t la **saisonnalité** (composante périodique) et R_t le **résidu** (composante aléatoire).

Modèle **multiplicatif** : $X_t = T_t \cdot S_t \cdot R_t$ (fréquent quand l'amplitude saisonnière croît avec le niveau).

1.4 Opérateur retard et différence

Définition 1.5 (Opérateur retard). L'opérateur retard B (ou *backshift operator*) est défini par :

$$BX_t = X_{t-1}, \quad B^k X_t = X_{t-k}.$$

L'opérateur différence est $\nabla = 1 - B : \nabla X_t = X_t - X_{t-1}$.

Remarque 1.6. On manipule les polynômes en B comme des polynômes ordinaires. Par exemple, $(1 - \phi B)X_t = \varepsilon_t$ s'écrit $X_t - \phi X_{t-1} = \varepsilon_t$.

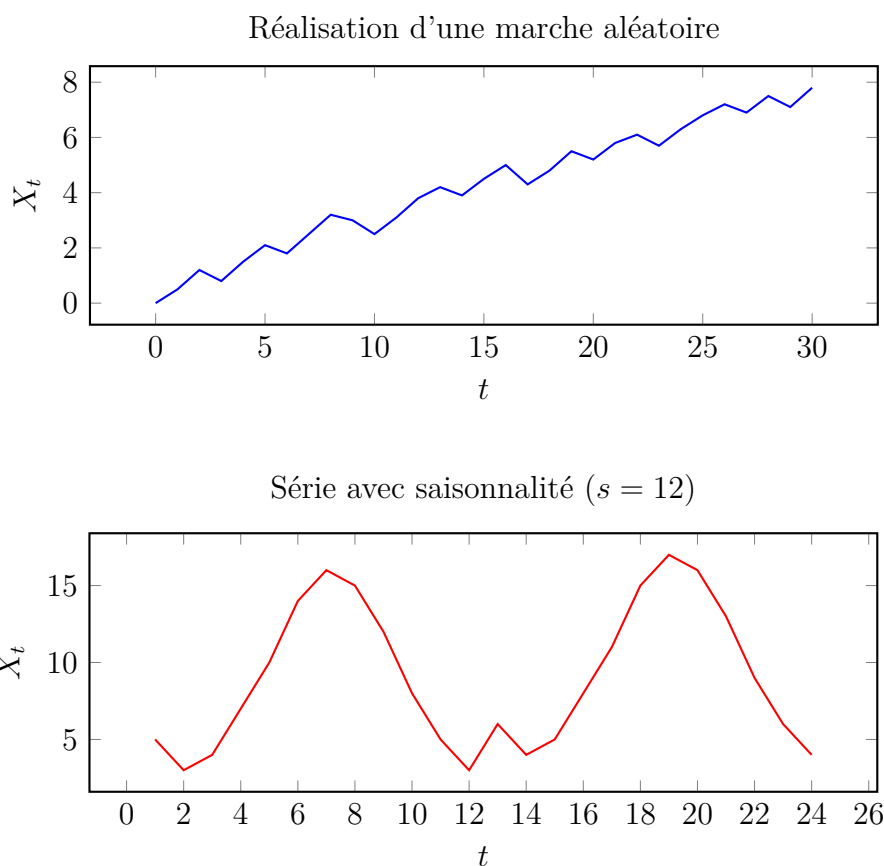
1.5 Exemples de séries temporelles

Exemple 1.7 (Marche aléatoire). $X_t = X_{t-1} + \varepsilon_t$ avec $\varepsilon_t \sim \text{BB}(0, \sigma^2)$. Alors $X_t = X_0 + \sum_{i=1}^t \varepsilon_i$ et $\text{Var}(X_t) = t\sigma^2 \rightarrow \infty$: la marche aléatoire n'est **pas stationnaire**.

Exemple 1.8 (Tendance linéaire + bruit). $X_t = a + bt + \varepsilon_t$. Pas stationnaire (la moyenne $\mathbb{E}[X_t] = a + bt$ dépend de t).

Exemple 1.9 (Série saisonnière). Températures mensuelles : période $s = 12$. On modélise S_t par des fonctions trigonométriques ou des indicatrices mensuelles.

1.6 Visualisation



1.7 Implémentation

Python

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

# Simuler une marche aléatoire
np.random.seed(42)
n = 500
eps = np.random.normal(0, 1, n)
rw = np.cumsum(eps)

# Simuler un AR(1)
phi = 0.8
ar1 = np.zeros(n)
for t in range(1, n):
    ar1[t] = phi * ar1[t-1] + eps[t]

# Charger des données réelles (exemple : AirPassengers)
# url =
# → 'https://raw.githubusercontent.com/jbrownlee/Datasets/master/airline-passengers.csv'
# df = pd.read_csv(url, parse_dates=['Month'], index_col='Month')

fig, axes = plt.subplots(2, 1, figsize=(12, 6))
axes[0].plot(rw, lw=0.8)
axes[0].set_title('Marche aléatoire')
axes[0].set_xlabel('t')

axes[1].plot(ar1, lw=0.8, color='red')
axes[1].set_title('AR(1) avec  $\phi=0.8$ ')
axes[1].set_xlabel('t')

plt.tight_layout()
plt.savefig('ch01_intro.pdf')
plt.show()
```

1.8 Exercices

Exercice 1.1 (*). Simuler 1000 réalisations d'une marche aléatoire de longueur $n = 200$. Tracer 10 trajectoires sur le même graphique. Calculer la variance empirique de X_{200} .

Exercice 1.2 (*). Charger la série `AirPassengers`. Identifier visuellement la tendance, la saisonnalité et la volatilité croissante. Proposer un modèle additif ou multiplicatif.

Exercice 1.3 (**). Montrer que si $X_t = X_{t-1} + \varepsilon_t$ (marche aléatoire), alors $\text{Var}(X_t) = t\sigma^2$ et $\text{Corr}(X_t, X_s) = \sqrt{\min(t, s) / \max(t, s)}$ pour $t, s > 0$.

Exercice 1.4 (***) – Projet). Télécharger les cours journaliers d'une action (ex. Apple) sur

5 ans. Décomposer la série en tendance, saisonnalité et résidu. Calculer les log-rendements $r_t = \ln(P_t/P_{t-1})$ et analyser leurs propriétés statistiques.

Formules clés

$$BX_t = X_{t-1}, \quad \nabla X_t = X_t - X_{t-1}$$

$$X_t = T_t + S_t + R_t \quad (\text{décomposition additive})$$

$$\text{Marche aléatoire : } X_t = X_{t-1} + \varepsilon_t, \quad \text{Var}(X_t) = t\sigma^2$$

Chapitre 2

Stationnarité et Autocovariance

2.1 Exemple motivant

La température journalière à Paris sur 30 ans présente une saisonnalité marquée mais, après désaisonnalisation, les résidus semblent fluctuer autour d'une moyenne constante avec une structure de dépendance invariante. C'est la **stationnarité**.

2.2 Stationnarité stricte et faible

Définition 2.1 (Stationnarité stricte). (X_t) est **strictement stationnaire** si pour tout $k \geq 1$, tout $(t_1, \dots, t_k)$ et tout $h \in \mathbb{Z}$:

$$(X_{t_1}, \dots, X_{t_k}) \stackrel{d}{=} (X_{t_1+h}, \dots, X_{t_k+h}).$$

Définition 2.2 (Stationnarité faible (au second ordre)). (X_t) est **faiblement stationnaire** (ou stationnaire au sens large) si :

1. $\mathbb{E}[X_t^2] < \infty$ pour tout t ,
2. $\mu = \mathbb{E}[X_t]$ ne dépend pas de t ,
3. $\gamma(t, t+h) = \text{Cov}(X_t, X_{t+h})$ ne dépend que de h .

Remarque 2.3. Stationnarité stricte + moments d'ordre 2 finis $\Rightarrow$ stationnarité faible. La réciproque est fautive en général, mais vraie pour les processus gaussiens.

2.3 Fonction d'autocovariance

Définition 2.4 (Autocovariance et autocorrélation). Pour un processus faiblement stationnaire :

$$\gamma(h) = \text{Cov}(X_t, X_{t+h}) = \mathbb{E}[(X_t - \mu)(X_{t+h} - \mu)], \quad (2.1)$$

$$\rho(h) = \frac{\gamma(h)}{\gamma(0)} = \text{Corr}(X_t, X_{t+h}). \quad (2.2)$$

La fonction $\rho(\cdot)$ est la **fonction d'autocorrélation** (ACF).

Théorème 2.5 (Propriétés de γ). 1. $\gamma(0) \geq 0$ (c'est la variance).

2. $\gamma(h) = \gamma(-h)$ (symétrie).
3. $|\gamma(h)| \leq \gamma(0)$ (par Cauchy-Schwarz).
4. γ est **semi-définie positive** : pour tout n , tout $(a_1, \dots, a_n) \in \mathbb{R}^n$ et tout $(t_1, \dots, t_n) :$

$$\sum_{i,j} a_i a_j \gamma(t_i - t_j) \geq 0.$$

Démonstration. (4) : $\sum_{i,j} a_i a_j \gamma(t_i - t_j) = \text{Var}(\sum_i a_i X_{t_i}) \geq 0.$ □

Théorème 2.6 (Herglotz). $\gamma : \mathbb{Z} \rightarrow \mathbb{R}$ est une fonction d'autocovariance si et seulement si elle est semi-définie positive. De plus, il existe une mesure positive finie F sur $[-\pi, \pi]$ (mesure spectrale) telle que :

$$\gamma(h) = \int_{-\pi}^{\pi} e^{i\omega h} dF(\omega).$$

2.4 Estimation de l'autocovariance

Définition 2.7 (Autocovariance empirique).

$$\hat{\gamma}(h) = \frac{1}{n} \sum_{t=1}^{n-|h|} (X_t - \bar{X})(X_{t+|h|} - \bar{X}), \quad \hat{\rho}(h) = \frac{\hat{\gamma}(h)}{\hat{\gamma}(0)}.$$

On divise par n (et non $n - |h|$) pour garantir que la matrice $(\hat{\gamma}(i - j))$ est semi-définie positive.

Théorème 2.8 (Bartlett). Pour un bruit blanc de taille n : $\hat{\rho}(h) \stackrel{\text{approx}}{\sim} \mathcal{N}(0, 1/n)$ pour $h \geq 1$. Les **bandes de Bartlett** $\pm 1,96/\sqrt{n}$ servent à tester si $\rho(h) = 0$.

2.5 Autocorrélation partielle

Définition 2.9 (PACF). L'autocorrélation partielle au décalage h est :

$$\alpha(h) = \text{Corr}(X_t - \hat{X}_t, X_{t+h} - \hat{X}_{t+h}),$$

où $\hat{X}_t$ et $\hat{X}_{t+h}$ sont les projections linéaires sur $(X_{t+1}, \dots, X_{t+h-1})$.

Équivalemment, $\alpha(h)$ est le dernier coefficient ϕ_{hh} dans la régression linéaire de X_t sur $(X_{t-1}, \dots, X_{t-h})$.

Théorème 2.10. Pour un $AR(p)$: $\alpha(h) = 0$ pour $h > p$. C'est l'outil principal pour identifier l'ordre p .

2.6 Tests de stationnarité

Définition 2.11 (Test de Dickey-Fuller augmenté (ADF)). Pour tester la présence d'une racine unitaire dans $X_t = \phi X_{t-1} + \varepsilon_t$:

$$H_0 : \phi = 1 \quad (\text{non stationnaire}) \quad \text{vs} \quad H_1 : |\phi| < 1 \quad (\text{stationnaire}).$$

On estime $\nabla X_t = (\phi - 1)X_{t-1} + \varepsilon_t$ et on teste si $\phi - 1 < 0$.

Attention

La distribution de la statistique ADF n'est **pas** gaussienne sous H_0 . On utilise les tables de Dickey–Fuller (ou les valeurs critiques calculées par simulation).

2.7 Implémentation

Python

```
import numpy as np
import matplotlib.pyplot as plt
from statsmodels.tsa.stattools import acf, pacf, adfuller
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf

# Simuler un AR(1)
np.random.seed(0)
n = 500
phi = 0.7
eps = np.random.normal(0, 1, n)
x = np.zeros(n)
for t in range(1, n):
    x[t] = phi * x[t-1] + eps[t]

# ACF et PACF
fig, axes = plt.subplots(1, 2, figsize=(12, 4))
plot_acf(x, lags=30, ax=axes[0], title='ACF')
plot_pacf(x, lags=30, ax=axes[1], title='PACF')
plt.tight_layout()
plt.savefig('ch02_acf_pacf.pdf')
plt.show()

# Test ADF
result = adfuller(x, regression='c')
print(f"ADF statistic: {result[0]:.4f}")
print(f"p-value: {result[1]:.4f}")
print(f"Stationnaire: {result[1] < 0.05}")

# Marche aléatoire (non stationnaire)
rw = np.cumsum(eps)
result_rw = adfuller(rw, regression='c')
print(f"\nMarche aléatoire - ADF: {result_rw[0]:.4f},  
↪ p={result_rw[1]:.4f}")
```

2.8 Exercices

Exercice 2.1 (*). Calculer la fonction d'autocovariance d'un bruit blanc $\varepsilon_t \sim \text{BB}(0, \sigma^2)$.

Exercice 2.2 (*). Calculer l'ACF de $X_t = \varepsilon_t + \theta\varepsilon_{t-1}$ (MA(1)). Montrer que $\rho(h) = 0$ pour $|h| \geq 2$.

Exercice 2.3 (★★). Montrer que si (X_t) est strictement stationnaire avec $\mathbb{E}[X_t^2] < \infty$, alors (X_t) est faiblement stationnaire.

Exercice 2.4 (★★). Démontrer que l'autocovariance empirique $\hat{\gamma}(h)$ (divisée par n) donne une matrice de Toeplitz semi-définie positive.

Exercice 2.5 (★★★ – Projet). Appliquer le test ADF et le test KPSS à 10 séries économiques (PIB, chômage, inflation, etc.). Comparer les conclusions et discuter les cas de désaccord.

Formules clés

$$\gamma(h) = \text{Cov}(X_t, X_{t+h}), \quad \rho(h) = \gamma(h)/\gamma(0)$$

$$\hat{\rho}(h) \stackrel{\text{approx}}{\sim} \mathcal{N}(0, 1/n) \quad \text{sous } H_0 : \rho(h) = 0$$

$$\text{ADF} : \nabla X_t = (\phi - 1)X_{t-1} + \varepsilon_t, \quad H_0 : \phi = 1$$

Chapitre 3

Modèles AR, MA, ARMA

3.1 Exemple motivant

Les log-rendements journaliers du CAC 40 montrent une corrélation faible mais significative aux premiers décalages. Un modèle AR(1) ou ARMA(1,1) capture cette dépendance linéaire.

3.2 Processus autorégressifs AR(p)

Définition 3.1 (AR(p)).

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \cdots + \phi_p X_{t-p} + \varepsilon_t,$$

où $\varepsilon_t \sim \text{BB}(0, \sigma^2)$. En notation opérateur : $\phi(B)X_t = \varepsilon_t$ avec $\phi(z) = 1 - \phi_1 z - \cdots - \phi_p z^p$.

Théorème 3.2 (Stationnarité d'un AR(p)). *L'AR(p) admet une unique solution stationnaire si et seulement si toutes les racines de $\phi(z) = 0$ sont **en dehors du disque unité** : $|z_i| > 1$ pour tout i .*

Esquisse pour AR(1). $X_t = \phi X_{t-1} + \varepsilon_t$. Par récurrence : $X_t = \sum_{j=0}^{\infty} \phi^j \varepsilon_{t-j}$. Cette série converge dans L^2 ssi $\sum \phi^{2j} < \infty$, i.e. $|\phi| < 1$. La racine de $\phi(z) = 1 - \phi z$ est $z = 1/\phi$, qui est hors du disque ssi $|\phi| < 1$. $\square$

Théorème 3.3 (ACF d'un AR(1)). *Si $X_t = \phi X_{t-1} + \varepsilon_t$ avec $|\phi| < 1$:*

$$\gamma(h) = \frac{\sigma^2 \phi^{|h|}}{1 - \phi^2}, \quad \rho(h) = \phi^{|h|}.$$

L'ACF décroît exponentiellement.

Démonstration. $\gamma(h) = \mathbb{E}[X_t X_{t+h}] = \phi \gamma(h-1)$ pour $h \geq 1$ (par multiplication de l'équation AR par X_{t+h} et prise d'espérance). Avec $\gamma(0) = \sigma^2/(1 - \phi^2)$. $\square$

Théorème 3.4 (PACF d'un AR(p)). *Pour un AR(p) : $\alpha(h) = 0$ pour tout $h > p$. La PACF tronque à l'ordre p .*

3.3 Équations de Yule–Walker

Définition 3.5. Pour un $\text{AR}(p)$ stationnaire, les relations :

$$\gamma(h) = \phi_1 \gamma(h-1) + \cdots + \phi_p \gamma(h-p) \quad (h \geq 1)$$

donnent le système de **Yule–Walker** :

$$\begin{pmatrix} \gamma(0) & \gamma(1) & \cdots & \gamma(p-1) \\ \gamma(1) & \gamma(0) & \cdots & \gamma(p-2) \\ \vdots & & \ddots & \vdots \\ \gamma(p-1) & \gamma(p-2) & \cdots & \gamma(0) \end{pmatrix} \begin{pmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_p \end{pmatrix} = \begin{pmatrix} \gamma(1) \\ \gamma(2) \\ \vdots \\ \gamma(p) \end{pmatrix}.$$

3.4 Processus moyenne mobile $\text{MA}(q)$

Définition 3.6 ($\text{MA}(q)$).

$$X_t = \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} = \theta(B) \varepsilon_t,$$

où $\theta(z) = 1 + \theta_1 z + \cdots + \theta_q z^q$.

Théorème 3.7 (Stationnarité d'un $\text{MA}(q)$). *Tout $\text{MA}(q)$ est **toujours stationnaire** (somme finie de bruits blancs).*

Théorème 3.8 (ACF d'un $\text{MA}(q)$).

$$\gamma(h) = \begin{cases} \sigma^2 \sum_{j=0}^{q-|h|} \theta_j \theta_{j+|h|} & \text{si } |h| \leq q, \\ 0 & \text{si } |h| > q, \end{cases}$$

avec $\theta_0 = 1$. L'ACF **tronque** au décalage q .

Définition 3.9 (Inversibilité). Un $\text{MA}(q)$ est **inversible** si toutes les racines de $\theta(z) = 0$ sont en dehors du disque unité. Dans ce cas, X_t admet une représentation $\text{AR}(\infty)$: $\pi(B)X_t = \varepsilon_t$ avec $\pi(z) = \theta(z)^{-1}$.

3.5 Processus $\text{ARMA}(p, q)$

Définition 3.10 ($\text{ARMA}(p, q)$).

$$\phi(B)X_t = \theta(B)\varepsilon_t, \quad \text{i.e.} \quad X_t - \sum_{i=1}^p \phi_i X_{t-i} = \varepsilon_t + \sum_{j=1}^q \theta_j \varepsilon_{t-j}.$$

Théorème 3.11 (Conditions de stationnarité et inversibilité). • *Stationnaire $\iff$ racines de $\phi(z) = 0$ hors du disque unité.*

- *Inversible $\iff$ racines de $\theta(z) = 0$ hors du disque unité.*
- *On exige toujours que $\phi(z)$ et $\theta(z)$ n'aient **pas de racine commune** (sinon le modèle est sur-paramétré).*

Théorème 3.12 (Représentation $\text{MA}(\infty)$). *Si l' $\text{ARMA}(p, q)$ est stationnaire : $X_t = \psi(B)\varepsilon_t$ avec $\psi(z) = \theta(z)/\phi(z) = \sum_{j=0}^{\infty} \psi_j z^j$. Les ψ_j sont les **poids de la réponse impulsionnelle**.*

3.6 Signatures ACF/PACF

Modèle	ACF	PACF
AR(p)	Décroissance exponentielle/sinusoïdale	Tronque après p
MA(q)	Tronque après q	Décroissance exponentielle/sinusoïdale
ARMA(p, q)	Décroissance après q	Décroissance après p

3.7 Implémentation

Python

```
import numpy as np
import matplotlib.pyplot as plt
from statsmodels.tsa.arima_process import ArmaProcess
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf

# Simuler un AR(2)
np.random.seed(42)
ar_params = np.array([1, -0.75, 0.25]) # 1 - 0.75*B + 0.25*B^2
ma_params = np.array([1])
ar2 = ArmaProcess(ar_params, ma_params)
x = ar2.generate_sample(nsample=500)

fig, axes = plt.subplots(1, 3, figsize=(15, 4))
axes[0].plot(x, lw=0.7)
axes[0].set_title('AR(2):  $\phi_1=0.75, \phi_2=-0.25$ ')
plot_acf(x, lags=25, ax=axes[1], title='ACF')
plot_pacf(x, lags=25, ax=axes[2], title='PACF')
plt.tight_layout()
plt.savefig('ch03_arma.pdf')
plt.show()

# Simuler un MA(2)
ma_params = np.array([1, 0.6, -0.3])
ma2 = ArmaProcess(np.array([1]), ma_params)
y = ma2.generate_sample(nsample=500)

# Simuler un ARMA(1,1)
arma11 = ArmaProcess(np.array([1, -0.8]), np.array([1, 0.4]))
z = arma11.generate_sample(nsample=500)

# Yule-Walker
from statsmodels.regression.linear_model import yule_walker
rho, sigma = yule_walker(x, order=2)
print(f"Yule-Walker: phi = {rho}")
```

3.8 Exercices

Exercice 3.1 (★). Calculer l'ACF d'un MA(1) : $X_t = \varepsilon_t + 0,5\varepsilon_{t-1}$. Vérifier numériquement.

Exercice 3.2 (★★). Montrer que l'AR(1) $X_t = \phi X_{t-1} + \varepsilon_t$ avec $|\phi| < 1$ a une représentation MA(∞) : $X_t = \sum_{j=0}^{\infty} \phi^j \varepsilon_{t-j}$.

Exercice 3.3 (★★). Montrer qu'un ARMA(1,1) avec $\phi = \theta$ se réduit à un bruit blanc (racine commune).

Exercice 3.4 (★★). Dériver les équations de Yule–Walker pour un AR(2) et résoudre le système.

Exercice 3.5 (★★★ – Projet). Simuler des processus AR(1), AR(2), MA(1), MA(2), ARMA(1,1) et ARMA(2,1). Pour chacun, tracer l'ACF et la PACF théoriques et empiriques. Construire un guide visuel d'identification.

Formules clés

AR(p) : $\phi(B)X_t = \varepsilon_t$, stationnaire ssi racines de $\phi(z)$ hors disque

MA(q) : $X_t = \theta(B)\varepsilon_t$, toujours stationnaire

ACF AR(1) : $\rho(h) = \phi^{|h|}$

ACF MA(q) : $\rho(h) = 0$ pour $|h| > q$

Chapitre 4

Identification, Estimation et Diagnostic

4.1 Exemple motivant

Face à une série temporelle réelle, comment choisir les ordres p et q d'un $\text{ARMA}(p, q)$, estimer les paramètres et vérifier que le modèle est adéquat ? C'est la **méthodologie de Box–Jenkins**.

4.2 La méthodologie de Box–Jenkins

Méthode

1. **Identification** : examiner l'ACF et la PACF pour proposer des ordres (p, q) .
2. **Estimation** : estimer $(\phi_1, \dots, \phi_p, \theta_1, \dots, \theta_q, \sigma^2)$ par maximum de vraisemblance ou moindres carrés conditionnels.
3. **Diagnostic** : vérifier que les résidus sont un bruit blanc.
4. **Prévision** : utiliser le modèle validé pour prévoir.

4.3 Identification

4.3.1 Examen de l'ACF et de la PACF

Règles pratiques (récapitulées au chapitre 3) :

- ACF tronque à $q \Rightarrow \text{MA}(q)$.
- PACF tronque à $p \Rightarrow \text{AR}(p)$.
- Les deux décroissent $\Rightarrow \text{ARMA}(p, q)$.

4.3.2 Critères d'information

Définition 4.1 (AIC et BIC).

$$\text{AIC}(p, q) = -2 \ln L(\hat{\theta}) + 2(p + q + 1), \quad (4.1)$$

$$\text{BIC}(p, q) = -2 \ln L(\hat{\theta}) + (p + q + 1) \ln n. \quad (4.2)$$

On choisit le modèle qui **minimise** le critère. Le BIC pénalise davantage la complexité.

Théorème 4.2 (Propriétés asymptotiques). • *L'AIC est asymptotiquement équivalent à la validation croisée leave-one-out.*

- *Le BIC est **consistant** : il sélectionne le vrai modèle avec probabilité $\rightarrow 1$.*
- *L'AIC tend à sur-paramétrer.*

4.4 Estimation

4.4.1 Maximum de vraisemblance

Définition 4.3. Pour un ARMA(p, q) gaussien, la log-vraisemblance exacte est :

$$\ln L(\theta) = -\frac{n}{2} \ln(2\pi) - \frac{1}{2} \sum_{t=1}^n \ln v_t - \frac{1}{2} \sum_{t=1}^n \frac{e_t^2}{v_t},$$

où $e_t = X_t - \hat{X}_{t|t-1}$ est l'erreur de prévision un pas en avant et $v_t = \text{Var}(e_t)$, calculés récursivement (algorithme des innovations ou filtre de Kalman).

Théorème 4.4 (Propriétés de l'EMV). *Sous des conditions de régularité, l'EMV $\hat{\theta}_n$ est :*

- *Consistant : $\hat{\theta}_n \xrightarrow{P} \theta_0$.*
- *Asymptotiquement normal : $\sqrt{n}(\hat{\theta}_n - \theta_0) \xrightarrow{d} \mathcal{N}(0, I(\theta_0)^{-1})$.*
- *Asymptotiquement efficace.*

4.4.2 Moindres carrés conditionnels

Définition 4.5. En conditionnant sur les premières observations, on minimise :

$$S(\theta) = \sum_{t=p+1}^n (X_t - \phi_1 X_{t-1} - \dots - \phi_p X_{t-p} - \theta_1 \hat{e}_{t-1} - \dots - \theta_q \hat{e}_{t-q})^2.$$

4.5 Diagnostic

4.5.1 Analyse des résidus

Définition 4.6. Les résidus du modèle sont $\hat{e}_t = X_t - \hat{X}_{t|t-1}$. Si le modèle est correct, les $\hat{e}_t$ doivent se comporter comme un bruit blanc.

Vérifications :

1. **ACF des résidus** : aucune autocorrélation significative.
2. **Test de Ljung–Box** :

$$Q(m) = n(n+2) \sum_{h=1}^m \frac{\hat{\rho}_{\varepsilon}(h)^2}{n-h} \stackrel{H_0}{\sim} \chi^2(m-p-q).$$

3. **Normalité** : QQ-plot, test de Jarque–Bera.
4. **Hétéroscédasticité** : effet ARCH (cf. chapitre 6).

Théorème 4.7 (Ljung–Box). *Sous H_0 (résidus = bruit blanc), $Q(m) \sim \chi^2(m-p-q)$ asymptotiquement. On rejette H_0 si $Q(m) > \chi^2_{1-\alpha}(m-p-q)$.*

4.6 Implémentation

Python

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.arima.model import ARIMA
from statsmodels.stats.diagnostic import acorr_ljungbox
from statsmodels.graphics.tsaplots import plot_acf

# Simuler un ARMA(1,1) connu
np.random.seed(42)
from statsmodels.tsa.arima_process import ArmaProcess
true_ar = np.array([1, -0.7])
true_ma = np.array([1, 0.3])
process = ArmaProcess(true_ar, true_ma)
data = process.generate_sample(nsample=500)

# Étape 1 : Identification (ACF/PACF déjà examinée)

# Étape 2 : Estimation par MLE
model = ARIMA(data, order=(1, 0, 1))
results = model.fit()
print(results.summary())
print(f"\nphi_1 = {results.arparams[0]:.4f} (vrai: 0.7)")
print(f"\ntheta_1 = {results.maparams[0]:.4f} (vrai: 0.3)")

# Étape 3 : Diagnostic
residuals = results.resid

fig, axes = plt.subplots(2, 2, figsize=(12, 8))
axes[0,0].plot(residuals, lw=0.5)
axes[0,0].set_title('Résidus')
plot_acf(residuals, lags=25, ax=axes[0,1], title='ACF des résidus')
axes[1,0].hist(residuals, bins=30, density=True, alpha=0.7)
axes[1,0].set_title('Distribution des résidus')
```

```

from scipy import stats
stats.probplot(residuals, dist='norm', plot=axes[1,1])
axes[1,1].set_title('QQ-plot')
plt.tight_layout()
plt.savefig('ch04_diagnostic.pdf')
plt.show()

# Test de Ljung-Box
lb_test = acorr_ljungbox(residuals, lags=20, return_df=True)
print(lb_test)

# Sélection par AIC/BIC
results_table = []
for p in range(4):
    for q in range(4):
        try:
            m = ARIMA(data, order=(p, 0, q)).fit()
            results_table.append({'p': p, 'q': q,
                                'AIC': m.aic, 'BIC': m.bic})
        except:
            pass
df = pd.DataFrame(results_table).sort_values('BIC')
print(df.head(5))

```

4.7 Exercices

Exercice 4.1 (★). Simuler un AR(2) et appliquer la méthodologie complète : identification, estimation, diagnostic.

Exercice 4.2 (★★). Montrer que le test de Ljung–Box avec m décalages suit asymptotiquement une $\chi^2(m - p - q)$ sous H_0 .

Exercice 4.3 (★★). Comparer AIC et BIC sur 1000 simulations d'un AR(1). Lequel sélectionne le bon ordre le plus souvent ? Le BIC sur-pénalise-t-il en petit échantillon ?

Exercice 4.4 (★★★ – Projet). Appliquer la méthodologie Box–Jenkins complète à la série du taux de chômage mensuel français. Présenter chaque étape avec justification.

Formules clés

$$\begin{aligned}
 \text{AIC} &= -2 \ln L + 2k, & \text{BIC} &= -2 \ln L + k \ln n \\
 Q(m) &= n(n+2) \sum_{h=1}^m \frac{\hat{\rho}_{\hat{\varepsilon}}(h)^2}{n-h} \sim \chi^2(m - p - q)
 \end{aligned}$$

Chapitre 5

Modèles ARIMA et Différenciation

5.1 Exemple motivant

Le PIB trimestriel français présente une tendance croissante : il n'est pas stationnaire. Après une différenciation ($\nabla X_t = X_t - X_{t-1}$), le taux de croissance est approximativement stationnaire et modélisable par un ARMA.

5.2 Processus intégrés

Définition 5.1 (Processus intégré d'ordre d). (X_t) est **intégré d'ordre d** , noté $X_t \sim I(d)$, si $\nabla^d X_t = (1 - B)^d X_t$ est stationnaire mais $\nabla^{d-1} X_t$ ne l'est pas.

Exemple 5.2. La marche aléatoire $X_t = X_{t-1} + \varepsilon_t$ est $I(1)$ car $\nabla X_t = \varepsilon_t$ est stationnaire.

5.3 Modèle ARIMA(p, d, q)

Définition 5.3 (ARIMA(p, d, q)).

$$\phi(B)(1 - B)^d X_t = \theta(B)\varepsilon_t.$$

X_t suit un ARIMA(p, d, q) si $W_t = \nabla^d X_t$ suit un ARMA(p, q).

Attention

On différencie pour rendre stationnaire, mais **trop différencier** (d trop grand) introduit des racines MA proches du cercle unité, ce qui dégrade les prévisions. En pratique, $d \leq 2$.

5.4 SARIMA : saisonnalité

Définition 5.4 (SARIMA(p, d, q) $\times$ (P, D, Q) $_s$). Pour une série de période s :

$$\Phi(B^s)\phi(B)(1 - B)^d(1 - B^s)^D X_t = \Theta(B^s)\theta(B)\varepsilon_t,$$

où Φ, Θ sont les polynômes saisonniers d'ordres P, Q .

Exemple 5.5 (Airline model). Le modèle de Box-Jenkins pour AirPassengers est un SARIMA(0, 1, 1) $\times$ (0, 1, 1) $_{12}$:

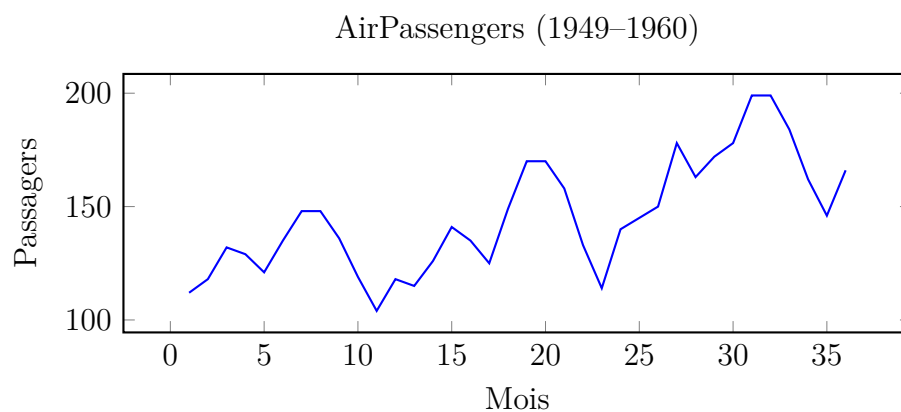
$$(1 - B)(1 - B^{12})X_t = (1 + \theta_1 B)(1 + \Theta_1 B^{12})\varepsilon_t.$$

5.5 Stratégie de différenciation

Méthode

1. Tracer la série et son ACF. Si l'ACF décroît très lentement, différencier.
2. Appliquer le test ADF (ou KPSS). Si H_0 (racine unitaire) n'est pas rejetée, différencier.
3. Après différenciation, vérifier que la série résultante est stationnaire.
4. Pour la saisonnalité, différencier à l'ordre s si l'ACF montre des pics périodiques.

5.6 Exemple numérique : AirPassengers



5.7 Implémentation

Python

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.arima.model import ARIMA
from statsmodels.tsa.statespace.sarimax import SARIMAX
from statsmodels.tsa.stattools import adfuller
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
import pmdarima as pm

# Charger AirPassengers
url = ('https://raw.githubusercontent.com/jbrownlee/Datasets/'
      'master/airline-passengers.csv')
df = pd.read_csv(url, parse_dates=['Month'], index_col='Month')
y = df['Passengers']

# Log-transformation (stabilise la variance)
```

```

y_log = np.log(y)

# Test ADF
print("ADF sur log(y):", adfuller(y_log)[1]) # > 0.05 => non stat.
y_diff = y_log.diff().dropna()
print("ADF sur diff(log(y)):", adfuller(y_diff)[1]) # < 0.05 => stat.

# SARIMA automatique
auto_model = pm.auto_arima(y_log, seasonal=True, m=12,
                           stepwise=True, trace=True)
print(auto_model.summary())

# Ajustement SARIMA(0,1,1)(0,1,1)[12]
model = SARIMAX(y_log, order=(0,1,1),
                seasonal_order=(0,1,1,12))
results = model.fit(dispatch=False)
print(results.summary())

# Pr vision
forecast = results.get_forecast(steps=24)
pred = np.exp(forecast.predicted_mean)
ci = np.exp(forecast.conf_int())

fig, ax = plt.subplots(figsize=(12, 5))
y.plot(ax=ax, label='Observ ')
pred.plot(ax=ax, label='Pr vision', color='red')
ax.fill_between(ci.index, ci.iloc[:,0], ci.iloc[:,1],
               alpha=0.2, color='red')

ax.legend()
ax.set_title('SARIMA - AirPassengers')
plt.savefig('ch05_arima.pdf')
plt.show()

```

5.8 Exercices

Exercice 5.1 (*). Montrer que $(1 - B)^2 X_t = X_t - 2X_{t-1} + X_{t-2}$.

Exercice 5.2 (*). Appliquer `pm.auto_arima` à une série de votre choix. Interpréter le modèle sélectionné.

Exercice 5.3 (**). Montrer que si $X_t \sim \text{ARIMA}(p, 1, q)$, alors les prévisions à long terme convergent vers une droite (prévision constante en niveaux différenciés).

Exercice 5.4 (*** – Projet). Modéliser le PIB trimestriel d'un pays par un ARIMA saisonnier. Comparer les prévisions hors échantillon avec différents ordres.

Formules clés

$$\text{ARIMA}(p, d, q) : \phi(B)(1 - B)^d X_t = \theta(B)\varepsilon_t$$

$$\text{SARIMA} : \Phi(B^s)\phi(B)(1 - B)^d(1 - B^s)^D X_t = \Theta(B^s)\theta(B)\varepsilon_t$$

$$I(d) : \nabla^d X_t \text{ stationnaire, } \nabla^{d-1} X_t \text{ non}$$

Chapitre 6

Modèles GARCH et Volatilité Conditionnelle

6.1 Exemple motivant

Les rendements financiers sont approximativement non corrélés, mais leur carré r_t^2 montre une forte autocorrélation : les périodes de forte volatilité sont groupées (**volatility clustering**). Les modèles ARMA ne capturent pas ce phénomène.

6.2 Faits stylisés des rendements financiers

1. Rendements approximativement non corrélés : $\rho_r(h) \approx 0$.
2. Rendements au carré corrélés : $\rho_{r^2}(h) > 0$ (persistance de la volatilité).
3. Queues lourdes (leptokurtose) : kurtosis > 3 .
4. Asymétrie de la volatilité (effet de levier) : les baisses augmentent davantage la volatilité que les hausses.

6.3 Modèle ARCH

Définition 6.1 (ARCH(q)).

$$r_t = \sigma_t z_t, \quad z_t \stackrel{\text{iid}}{\sim} (0, 1), \quad \sigma_t^2 = \omega + \sum_{i=1}^q \alpha_i r_{t-i}^2,$$

où $\omega > 0$, $\alpha_i \geq 0$, et $\sum \alpha_i < 1$ pour la stationnarité.

Théorème 6.2 (Propriétés d'un ARCH(q)). 1. $\mathbb{E}[r_t] = 0$, $\text{Var}(r_t) = \omega/(1 - \sum \alpha_i)$.

2. (r_t) est un bruit blanc (non corrélé).
3. (r_t^2) suit un AR(q) : $r_t^2 = \omega + \sum \alpha_i r_{t-i}^2 + v_t$ où $v_t = r_t^2 - \sigma_t^2$ est un bruit blanc.
4. Kurtosis > 3 (queues lourdes même si z_t est gaussien).

6.4 Modèle GARCH

Définition 6.3 (GARCH(p, q)).

$$r_t = \sigma_t z_t, \quad \sigma_t^2 = \omega + \sum_{i=1}^q \alpha_i r_{t-i}^2 + \sum_{j=1}^p \beta_j \sigma_{t-j}^2.$$

Conditions de stationnarité : $\omega > 0$, $\alpha_i, \beta_j \geq 0$, $\sum_{i=1}^q \alpha_i + \sum_{j=1}^p \beta_j < 1$.

Théorème 6.4 (Variance inconditionnelle GARCH(1, 1)).

$$\text{Var}(r_t) = \frac{\omega}{1 - \alpha_1 - \beta_1}.$$

La *persistance* de la volatilité est mesurée par $\alpha_1 + \beta_1$ (proche de 1 en finance).

Intuition

Le GARCH(1, 1) est le modèle de volatilité le plus utilisé en pratique. La prévision de la variance conditionnelle un pas en avant est :

$$\hat{\sigma}_{t+1}^2 = \omega + \alpha_1 r_t^2 + \beta_1 \sigma_t^2.$$

C'est une moyenne pondérée exponentielle des carrés passés.

6.5 IGARCH

Définition 6.5. Si $\alpha_1 + \beta_1 = 1$ (intégré GARCH), les chocs de volatilité persistent indéfiniment. Le processus n'est pas stationnaire au second ordre mais reste **strictement stationnaire**.

6.6 Extensions

6.6.1 EGARCH

Définition 6.6. Le modèle EGARCH de Nelson :

$$\ln \sigma_t^2 = \omega + \alpha(|z_{t-1}| - \mathbb{E}[|z_{t-1}|]) + \gamma z_{t-1} + \beta \ln \sigma_{t-1}^2.$$

Le terme γz_{t-1} capture l'**effet de levier**. Pas de contrainte de positivité sur les paramètres.

6.6.2 GJR-GARCH

Définition 6.7.

$$\sigma_t^2 = \omega + (\alpha + \gamma \mathbf{1}_{r_{t-1} < 0}) r_{t-1}^2 + \beta \sigma_{t-1}^2.$$

$\gamma > 0$ implique que les rendements négatifs augmentent davantage la volatilité.

6.7 Estimation

Définition 6.8 (Quasi-maximum de vraisemblance). La log-vraisemblance conditionnelle gaussienne :

$$\ell(\boldsymbol{\theta}) = -\frac{1}{2} \sum_{t=1}^n \left(\ln \sigma_t^2 + \frac{r_t^2}{\sigma_t^2} \right).$$

Même si z_t n'est pas gaussien, le QMLE est consistant sous certaines conditions.

6.8 Tests d'effet ARCH

Définition 6.9 (Test ARCH-LM d'Engle). Régresser r_t^2 sur $r_{t-1}^2, \dots, r_{t-q}^2$ et tester la nullité jointe par $nR^2 \sim \chi^2(q)$.

6.9 Implémentation

Python

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from arch import arch_model

# Simuler un GARCH(1,1)
np.random.seed(42)
n = 1000
omega, alpha, beta = 0.05, 0.10, 0.85
sigma2 = np.zeros(n)
r = np.zeros(n)
sigma2[0] = omega / (1 - alpha - beta)
for t in range(1, n):
    sigma2[t] = omega + alpha * r[t-1]**2 + beta * sigma2[t-1]
    r[t] = np.sqrt(sigma2[t]) * np.random.normal()

fig, axes = plt.subplots(3, 1, figsize=(12, 8))
axes[0].plot(r, lw=0.5)
axes[0].set_title('Rendements simulés (GARCH(1,1))')
axes[1].plot(r**2, lw=0.5, color='orange')
axes[1].set_title('$r_t^2$ (volatilité réalisée)')
axes[2].plot(np.sqrt(sigma2), lw=0.5, color='red')
axes[2].set_title('$\sigma_t$ (volatilité conditionnelle)')
plt.tight_layout()
plt.savefig('ch06_garch.pdf')
plt.show()

# Ajuster un GARCH(1,1)
am = arch_model(r, vol='Garch', p=1, q=1, dist='normal')
res = am.fit(displ='off')
print(res.summary())
```

```
# Prédiction de volatilité
forecasts = res.forecast(horizon=10)
print("Variance prévisionnelle:")
print(forecasts.variance.iloc[-1])
```

6.10 Exercices

Exercice 6.1 (*). Montrer qu'un ARCH(1) avec $r_t = \sigma_t z_t$, $\sigma_t^2 = \omega + \alpha r_{t-1}^2$ implique $\mathbb{E}[r_t^4] = 3\omega^2(1 + \alpha)/(1 - 3\alpha^2)$ (si $3\alpha^2 < 1$). En déduire la kurtosis.

Exercice 6.2 (**). Montrer que le GARCH(1, 1) peut s'écrire comme un ARCH(∞).

Exercice 6.3 (**). Démontrer que le processus r_t^2 d'un GARCH(1, 1) suit un ARMA(1, 1).

Exercice 6.4 (***) – Projet). Télécharger les rendements journaliers du S&P 500 sur 10 ans. Estimer un GARCH(1, 1), un EGARCH(1, 1) et un GJR-GARCH(1, 1). Comparer par AIC et analyser l'effet de levier.

Formules clés

$$\text{GARCH}(1, 1) : \sigma_t^2 = \omega + \alpha r_{t-1}^2 + \beta \sigma_{t-1}^2, \quad \alpha + \beta < 1$$

$$\text{Var}(r_t) = \omega / (1 - \alpha - \beta)$$

$$\text{EGARCH} : \ln \sigma_t^2 = \omega + \alpha(|z_{t-1}| - \mathbb{E}|z_{t-1}|) + \gamma z_{t-1} + \beta \ln \sigma_{t-1}^2$$

Chapitre 7

Analyse Spectrale

7.1 Exemple motivant

Un signal climatique contient des cycles de différentes périodes (journalier, annuel, El Niño). L'analyse spectrale décompose la variance totale par fréquence, révélant les périodicités cachées.

7.2 Densité spectrale

Définition 7.1 (Densité spectrale de puissance). Si $\sum_{h=-\infty}^{\infty} |\gamma(h)| < \infty$, la **densité spectrale** de (X_t) est :

$$f(\omega) = \frac{1}{2\pi} \sum_{h=-\infty}^{\infty} \gamma(h) e^{-i\omega h}, \quad \omega \in [-\pi, \pi].$$

C'est la transformée de Fourier de la fonction d'autocovariance.

Théorème 7.2 (Propriétés de f). 1. $f(\omega) \geq 0$ pour tout ω .

2. f est paire : $f(\omega) = f(-\omega)$.

3. Relation inverse : $\gamma(h) = \int_{-\pi}^{\pi} f(\omega) e^{i\omega h} d\omega$.

4. $\gamma(0) = \text{Var}(X_t) = \int_{-\pi}^{\pi} f(\omega) d\omega$ (théorème de Parseval).

Esquisse de (1). Par le théorème de Herglotz, γ est semi-définie positive. Donc $f(\omega) = \frac{1}{2\pi} \sum \gamma(h) e^{-i\omega h} \geq 0$ (comme limite de formes quadratiques positives). $\square$

7.3 Spectres de processus classiques

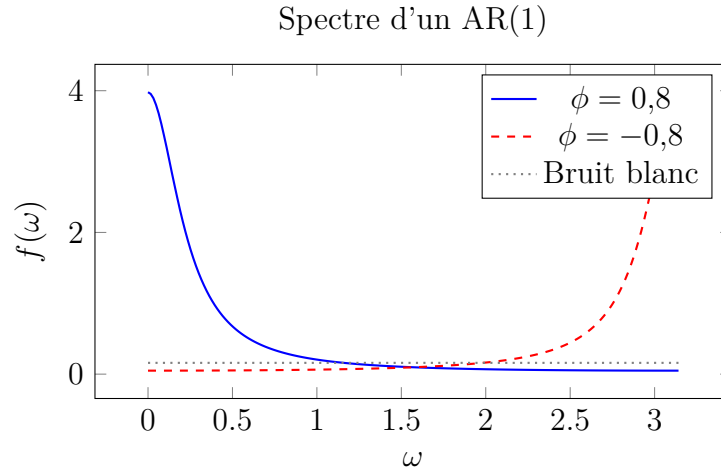
Théorème 7.3 (Spectre d'un ARMA(p, q)).

$$f(\omega) = \frac{\sigma^2}{2\pi} \frac{|\theta(e^{-i\omega})|^2}{|\phi(e^{-i\omega})|^2}.$$

Exemple 7.4 (Spectre d'un AR(1)). $\phi(z) = 1 - \phi z$:

$$f(\omega) = \frac{\sigma^2}{2\pi} \frac{1}{|1 - \phi e^{-i\omega}|^2} = \frac{\sigma^2}{2\pi(1 - 2\phi \cos \omega + \phi^2)}.$$

Si $\phi > 0$: pic en $\omega = 0$ (basses fréquences dominantes). Si $\phi < 0$: pic en $\omega = \pi$.



7.4 Périodogramme

Définition 7.5 (Périodogramme). L'estimateur naïf de $f(\omega)$:

$$I_n(\omega) = \frac{1}{2\pi n} \left| \sum_{t=1}^n X_t e^{-i\omega t} \right|^2.$$

Théorème 7.6 (Propriétés du périodogramme). 1. $\mathbb{E}[I_n(\omega_j)] \rightarrow f(\omega_j)$ pour les fréquences de Fourier $\omega_j = 2\pi j/n$.

2. $I_n(\omega_j)$ est un estimateur **non consistant** : $\text{Var}(I_n(\omega_j)) \not\rightarrow 0$.

3. Aux fréquences de Fourier : $I_n(\omega_j)/f(\omega_j) \stackrel{\text{approx}}{\sim} \chi^2(2)/2$.

Attention

Le périodogramme brut est très erratique. Il faut le lisser pour obtenir un estimateur consistant.

7.5 Estimation spectrale

7.5.1 Fenêtre spectrale (lissage)

Définition 7.7 (Périodogramme lissé).

$$\hat{f}(\omega) = \sum_{|h| \leq M} w(h) \hat{\gamma}(h) e^{-i\omega h},$$

où w est une **fenêtre** (Bartlett, Parzen, Tukey, etc.) et M est la **largeur de bande**.

7.5.2 Méthode paramétrique

Estimer un modèle AR(p) par Yule–Walker ou Burg, puis utiliser le spectre théorique $\hat{f}(\omega) = \frac{\hat{\sigma}^2}{2\pi|\hat{\phi}(e^{-i\omega})|^2}$.

7.6 Implémentation

Python

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.signal import periodogram, welch

# Simuler un AR(2) avec pic spectral
np.random.seed(42)
n = 1024
eps = np.random.normal(0, 1, n)
x = np.zeros(n)
phi1, phi2 = 1.5, -0.85 # racines complexes => pic
for t in range(2, n):
    x[t] = phi1*x[t-1] + phi2*x[t-2] + eps[t]

# Périodogramme brut
freq_p, Pxx_p = periodogram(x, fs=1, scaling='density')

# Périodogramme lissé (Welch)
freq_w, Pxx_w = welch(x, fs=1, nperseg=256, scaling='density')

fig, axes = plt.subplots(1, 2, figsize=(14, 5))
axes[0].semilogy(freq_p, Pxx_p, lw=0.5)
axes[0].set_title('Périodogramme brut')
axes[0].set_xlabel('Fréquence')

axes[1].semilogy(freq_w, Pxx_w, lw=1.5, color='red')
axes[1].set_title('Périodogramme lissé (Welch)')
axes[1].set_xlabel('Fréquence')

plt.tight_layout()
plt.savefig('ch07_spectral.pdf')
plt.show()

# Spectre théorique AR(2)
omega = np.linspace(0.01, np.pi, 500)
phi_z = 1 - phi1*np.exp(-1j*omega) - phi2*np.exp(-2j*omega)
f_theo = 1 / (2*np.pi * np.abs(phi_z)**2)

plt.figure(figsize=(8, 4))
plt.plot(omega, f_theo, 'b-', label='Théorique')
plt.title('Spectre théorique AR(2)')
plt.xlabel('$\\omega$'); plt.ylabel('$f(\\omega)$')
plt.legend()
plt.savefig('ch07_spectral_theo.pdf')
plt.show()
```

7.7 Exercices

Exercice 7.1 (*). Calculer la densité spectrale d'un MA(1) : $X_t = \varepsilon_t + \theta\varepsilon_{t-1}$.

Exercice 7.2 (**). Montrer que le périodogramme peut s'écrire $I_n(\omega) = \sum_{|h| < n} \hat{\gamma}(h) e^{-i\omega h} / (2\pi)$.

Exercice 7.3 (**). Démontrer que pour un bruit blanc gaussien, $2nI_n(\omega_j)/\sigma^2 \sim \chi^2(2)$ aux fréquences de Fourier non nulles.

Exercice 7.4 (*** – Projet). Analyser le spectre des températures journalières de Paris sur 50 ans. Identifier les pics correspondant aux cycles annuel, semi-annuel et hebdomadaire.

Formules clés

$$f(\omega) = \frac{1}{2\pi} \sum_h \gamma(h) e^{-i\omega h}, \quad \gamma(h) = \int_{-\pi}^{\pi} f(\omega) e^{i\omega h} d\omega$$

$$f_{\text{ARMA}}(\omega) = \frac{\sigma^2 |\theta(e^{-i\omega})|^2}{2\pi |\phi(e^{-i\omega})|^2}$$

$$I_n(\omega) = \frac{1}{2\pi n} \left| \sum_{t=1}^n X_t e^{-i\omega t} \right|^2$$

Chapitre 8

Modèles à Espace d'États et Filtre de Kalman

8.1 Exemple motivant

Le PIB « vrai » d'une économie est inobservable ; on n'observe que des estimations bruitées et révisées. Un modèle à espace d'états sépare l'état **latent** de l'**observation bruitée** et le filtre de Kalman fournit l'estimation optimale de l'état.

8.2 Modèle linéaire gaussien à espace d'états

Définition 8.1 (Modèle à espace d'états).

$$\mathbf{x}_{t+1} = F\mathbf{x}_t + G\mathbf{w}_t \quad (\text{équation d'état}) \quad (8.1)$$

$$\mathbf{y}_t = H\mathbf{x}_t + \mathbf{v}_t \quad (\text{équation d'observation}) \quad (8.2)$$

où $\mathbf{x}_t \in \mathbb{R}^p$ est l'état (latent), $\mathbf{y}_t \in \mathbb{R}^m$ l'observation, $\mathbf{w}_t \sim \mathcal{N}(0, Q)$ et $\mathbf{v}_t \sim \mathcal{N}(0, R)$ sont indépendants.

Exemple 8.2 (Tendance locale (local level)). $x_{t+1} = x_t + w_t$, $y_t = x_t + v_t$. L'état est la « vraie valeur » lissée.

Exemple 8.3 (ARMA comme espace d'états). Tout ARMA(p, q) peut se mettre sous forme espace d'états. Par exemple, l'AR(1) $X_t = \phi X_{t-1} + \varepsilon_t$ correspond à $F = \phi$, $H = 1$, $G = 1$, $Q = \sigma^2$, $R = 0$.

8.3 Filtre de Kalman

Définition 8.4 (Filtre de Kalman). Notations : $\hat{\mathbf{x}}_{t|s} = \mathbb{E}[\mathbf{x}_t | \mathbf{y}_1, \dots, \mathbf{y}_s]$ et $P_{t|s} = \text{Cov}(\mathbf{x}_t - \hat{\mathbf{x}}_{t|s})$.

Prédiction :

$$\hat{\mathbf{x}}_{t+1|t} = F\hat{\mathbf{x}}_{t|t}, \quad (8.3)$$

$$P_{t+1|t} = FP_{t|t}F^\top + GQG^\top. \quad (8.4)$$

Mise à jour :

$$K_t = P_{t|t-1} H^\top (H P_{t|t-1} H^\top + R)^{-1} \quad (\text{gain de Kalman}), \quad (8.5)$$

$$\hat{\mathbf{x}}_{t|t} = \hat{\mathbf{x}}_{t|t-1} + K_t (\mathbf{y}_t - H \hat{\mathbf{x}}_{t|t-1}), \quad (8.6)$$

$$P_{t|t} = (I - K_t H) P_{t|t-1}. \quad (8.7)$$

Théorème 8.5 (Optimalité du filtre de Kalman). *Dans le modèle linéaire gaussien, le filtre de Kalman donne l'estimateur MMSE (minimum mean square error) de $\mathbf{x}_t$ sachant $\mathbf{y}_1, \dots, \mathbf{y}_t$. C'est l'estimateur **optimal** parmi tous les estimateurs (pas seulement linéaires).*

Esquisse. Par récurrence. À chaque pas, la distribution conditionnelle $\mathbf{x}_t | \mathbf{y}_{1:t}$ est gaussienne (par fermeture de la loi gaussienne par conditionnement linéaire). L'espérance conditionnelle est l'estimateur MMSE pour les distributions gaussiennes. $\square$

8.4 Lisseur de Kalman

Définition 8.6 (Lisseur de Rauch–Tung–Striebel). Pour calculer $\hat{\mathbf{x}}_{t|n}$ (estimation de l'état passé avec toutes les observations) :

$$J_t = P_{t|t} F^\top P_{t+1|t}^{-1}, \quad (8.8)$$

$$\hat{\mathbf{x}}_{t|n} = \hat{\mathbf{x}}_{t|t} + J_t (\hat{\mathbf{x}}_{t+1|n} - \hat{\mathbf{x}}_{t+1|t}), \quad (8.9)$$

$$P_{t|n} = P_{t|t} + J_t (P_{t+1|n} - P_{t+1|t}) J_t^\top. \quad (8.10)$$

On itère en arrière de $t = n - 1$ à $t = 1$.

8.5 Vraisemblance par le filtre de Kalman

Théorème 8.7 (Décomposition de l'innovation). *La log-vraisemblance se décompose :*

$$\ln L = -\frac{nm}{2} \ln(2\pi) - \frac{1}{2} \sum_{t=1}^n [\ln |S_t| + \mathbf{e}_t^\top S_t^{-1} \mathbf{e}_t],$$

où $\mathbf{e}_t = \mathbf{y}_t - H \hat{\mathbf{x}}_{t|t-1}$ est l'**innovation** et $S_t = H P_{t|t-1} H^\top + R$.

8.6 Implémentation

Python

```
import numpy as np
import matplotlib.pyplot as plt
from statsmodels.tsa.statespace.structural import UnobservedComponents

# Simuler un modèle local level
np.random.seed(42)
n = 200
sigma_w, sigma_v = 0.5, 1.0
```

```
x = np.zeros(n)
y = np.zeros(n)
for t in range(1, n):
    x[t] = x[t-1] + sigma_w * np.random.normal()
    y[t] = x[t] + sigma_v * np.random.normal()

# Filtre de Kalman manuel
F, H, Q, R = 1.0, 1.0, sigma_w**2, sigma_v**2
x_filt = np.zeros(n)
P_filt = np.zeros(n)
P_filt[0] = 1.0
for t in range(1, n):
    # Prédiction
    x_pred = F * x_filt[t-1]
    P_pred = F * P_filt[t-1] * F + Q
    # Mise à jour
    S = H * P_pred * H + R
    K = P_pred * H / S
    x_filt[t] = x_pred + K * (y[t] - H * x_pred)
    P_filt[t] = (1 - K * H) * P_pred

fig, ax = plt.subplots(figsize=(12, 5))
ax.plot(y, 'o', ms=2, alpha=0.5, label='Observations $y_t$')
ax.plot(x, 'g-', lw=1.5, label='État vrai $x_t$')
ax.plot(x_filt, 'r--', lw=1.5, label='Filtre de Kalman $\hat{x}_{t|t}$')
ax.legend()
ax.set_title('Filtre de Kalman - Local Level Model')
plt.savefig('ch08_kalman.pdf')
plt.show()

# Avec statsmodels
model = UnobservedComponents(y, 'local level')
results = model.fit(dispatch=False)
print(results.summary())
```

8.7 Exercices

Exercice 8.1 (*). Écrire l'AR(2) sous forme espace d'états.

Exercice 8.2 (**). Démontrer que le gain de Kalman K_t minimise $\text{tr}(P_{t|t})$.

Exercice 8.3 (**). Montrer que dans le modèle local level stationnaire (σ_w, σ_v constants), le gain de Kalman converge vers une constante K_∞ .

Exercice 8.4 (***) – Projet). Implémenter un filtre de Kalman étendu (EKF) pour un modèle de volatilité stochastique : $\ln \sigma_t^2 = \phi \ln \sigma_{t-1}^2 + w_t$, $r_t = \sigma_t z_t$.

Formules clés

$$\hat{\mathbf{x}}_{t+1|t} = F\hat{\mathbf{x}}_{t|t}, \quad P_{t+1|t} = FP_{t|t}F^\top + GQG^\top$$

$$K_t = P_{t|t-1}H^\top (HP_{t|t-1}H^\top + R)^{-1}$$

$$\hat{\mathbf{x}}_{t|t} = \hat{\mathbf{x}}_{t|t-1} + K_t(\mathbf{y}_t - H\hat{\mathbf{x}}_{t|t-1})$$

$$\ln L = -\frac{1}{2} \sum [\ln |S_t| + \mathbf{e}_t^\top S_t^{-1} \mathbf{e}_t]$$

Chapitre 9

Séries Temporelles Multivariées — VAR et Cointégration

9.1 Exemple motivant

Le taux d'intérêt et l'inflation évoluent conjointement : la hausse de l'un influence l'autre. Un modèle **VAR** capture ces interdépendances dynamiques. La **coïntégration** révèle des relations d'équilibre long terme entre variables non stationnaires.

9.2 Processus vectoriels stationnaires

Définition 9.1 (Processus vectoriel stationnaire). $(\mathbf{Y}_t)_{t \in \mathbb{Z}}$ avec $\mathbf{Y}_t \in \mathbb{R}^k$ est **faiblement stationnaire** si :

1. $\mathbb{E}[\|\mathbf{Y}_t\|^2] < \infty$,
2. $\boldsymbol{\mu} = \mathbb{E}[\mathbf{Y}_t]$ ne dépend pas de t ,
3. $\Gamma(h) = \text{Cov}(\mathbf{Y}_t, \mathbf{Y}_{t+h}) = \mathbb{E}[(\mathbf{Y}_t - \boldsymbol{\mu})(\mathbf{Y}_{t+h} - \boldsymbol{\mu})^\top]$ ne dépend que de h .

Note : $\Gamma(h) = \Gamma(-h)^\top$ (mais $\Gamma(h) \neq \Gamma(h)^\top$ en général).

9.3 Modèle VAR(p)

Définition 9.2 (VAR(p)).

$$\mathbf{Y}_t = \mathbf{c} + A_1 \mathbf{Y}_{t-1} + A_2 \mathbf{Y}_{t-2} + \cdots + A_p \mathbf{Y}_{t-p} + \mathbf{u}_t,$$

où $\mathbf{u}_t \sim \text{BB}(\mathbf{0}, \Sigma_u)$ (k -dimensionnel) et $A_i \in \mathbb{R}^{k \times k}$.

Théorème 9.3 (Stationnarité d'un VAR(p)). *Le VAR(p) est stationnaire si et seulement si toutes les racines de*

$$\det(I_k - A_1 z - \cdots - A_p z^p) = 0$$

sont en dehors du disque unité.

9.3.1 Estimation

Théorème 9.4. *Les paramètres d'un VAR(p) peuvent être estimés équation par équation par OLS. L'estimateur est consistant et asymptotiquement normal.*

9.4 Causalité de Granger

Définition 9.5 (Causalité de Granger). $Y_{2,t}$ **cause au sens de Granger** $Y_{1,t}$ si les valeurs passées de Y_2 améliorent la prévision de Y_1 , i.e. si dans le VAR, les coefficients associés aux retards de Y_2 dans l'équation de Y_1 sont conjointement significatifs.

Test : $H_0 : A_1^{(12)} = \dots = A_p^{(12)} = 0$ par test de Wald (F -test).

9.5 Fonctions de réponse impulsionnelle

Définition 9.6 (IRF). La **fonction de réponse impulsionnelle** mesure l'effet d'un choc unitaire sur $u_{j,t}$ sur $Y_{i,t+h}$:

$$\Psi_h = \frac{\partial \mathbf{Y}_{t+h}}{\partial \mathbf{u}_t^\top},$$

où $\mathbf{Y}_t = \sum_{h=0}^{\infty} \Psi_h \mathbf{u}_{t-h}$ est la représentation MA(∞).

9.6 Coïntégration

Définition 9.7 (Coïntégration). Des séries $I(1)$ $Y_{1,t}, \dots, Y_{k,t}$ sont **coïntégrées** s'il existe $\beta \neq \mathbf{0}$ tel que $\beta^\top \mathbf{Y}_t \sim I(0)$. Le vecteur β est un **vecteur de coïntégration** et $\beta^\top \mathbf{Y}_t$ représente une relation d'équilibre long terme.

Exemple 9.8. Les prix spot et futures d'une matière première sont chacun $I(1)$, mais leur différence (la base) est $I(0)$: ils sont coïntégrés avec $\beta = (1, -1)^\top$.

Définition 9.9 (Modèle à correction d'erreur (VECM)).

$$\nabla \mathbf{Y}_t = \Pi \mathbf{Y}_{t-1} + \sum_{i=1}^{p-1} \Gamma_i \nabla \mathbf{Y}_{t-i} + \mathbf{u}_t,$$

où $\Pi = \alpha \beta^\top$ avec $\alpha \in \mathbb{R}^{k \times r}$ (vitesses d'ajustement) et $\beta \in \mathbb{R}^{k \times r}$ (vecteurs de coïntégration), $r = \text{rang}(\Pi)$ est le nombre de relations de coïntégration.

Théorème 9.10 (Représentation de Granger). *Si $\mathbf{Y}_t \sim I(1)$ et coïntégré de rang r :*

- $0 < r < k$: coïntégration. Le VECM est la bonne spécification.
- $r = 0$: pas de coïntégration, VAR en différences.
- $r = k$: toutes les séries sont $I(0)$, VAR en niveaux.

9.6.1 Test de Johansen

Définition 9.11. Le test de Johansen détermine le rang de coïntégration r par des tests séquentiels de la valeur propre maximale ou de la trace.

9.7 Implémentation

Python

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.api import VAR
from statsmodels.tsa.vector_ar.vecm import coint_johansen

# Simuler un VAR(1) bivarié
np.random.seed(42)
n = 500
A = np.array([[0.5, 0.3], [-0.2, 0.6]])
Y = np.zeros((n, 2))
for t in range(1, n):
    Y[t] = A @ Y[t-1] + np.random.multivariate_normal([0,0],
                                                         [[1, 0.3],[0.3, 1]])

df = pd.DataFrame(Y, columns=['Y1', 'Y2'])

# Ajuster VAR
model = VAR(df)
results = model.fit(maxlags=5, ic='bic')
print(results.summary())

# Causalité de Granger
granger = results.test_causality('Y1', 'Y2', kind='f')
print(f"\nGranger Y2 -> Y1: p-value = {granger.pvalue:.4f}")

# IRF
irf = results.irf(20)
irf.plot(orth=True)
plt.savefig('ch09_irf.pdf')
plt.show()

# Coïntégration (Johansen) sur données simulées I(1)
rw1 = np.cumsum(np.random.normal(0, 1, n))
rw2 = 0.5 * rw1 + np.cumsum(np.random.normal(0, 0.3, n))
data_coint = np.column_stack([rw1, rw2])
joh = coint_johansen(data_coint, det_order=0, k_ar_diff=1)
print("\nJohansen trace stats:", joh.lr1)
print("Critical values (90%, 95%, 99%):", joh.cvt)
```

9.8 Exercices

Exercice 9.1 (*). Vérifier que le VAR(1) $\mathbf{Y}_t = A\mathbf{Y}_{t-1} + \mathbf{u}_t$ avec $A = \begin{pmatrix} 0,5 & 0 \\ 0 & 0,3 \end{pmatrix}$ est stationnaire. Calculer $\Gamma(0)$.

Exercice 9.2 (**). Montrer que dans un VAR(1), $\Gamma(h) = A^h\Gamma(0)$ pour $h \geq 0$.

Exercice 9.3 (★★). Montrer que si $Y_{1,t}$ et $Y_{2,t}$ sont des marches aléatoires indépendantes, elles ne sont pas coïntégrées.

Exercice 9.4 (★★★ – Projet). Estimer un VAR pour le couple (taux d'intérêt, inflation) d'un pays. Tester la causalité de Granger dans les deux sens et interpréter les fonctions de réponse impulsionnelle.

Formules clés

$$\text{VAR}(p) : \mathbf{Y}_t = \mathbf{c} + A_1 \mathbf{Y}_{t-1} + \cdots + A_p \mathbf{Y}_{t-p} + \mathbf{u}_t$$

$$\text{VECM} : \nabla \mathbf{Y}_t = \alpha \beta^\top \mathbf{Y}_{t-1} + \sum \Gamma_i \nabla \mathbf{Y}_{t-i} + \mathbf{u}_t$$

$$\text{Coïntégration} : \beta^\top \mathbf{Y}_t \sim I(0) \text{ avec } \mathbf{Y}_t \sim I(1)$$

Chapitre 10

Prévision et Intervalles de Confiance

10.1 Exemple motivant

Prévoir la demande d'électricité pour le lendemain, avec un intervalle de confiance, est essentiel pour la gestion du réseau. La qualité de la prévision détermine les coûts économiques.

10.2 Prévision optimale

Définition 10.1 (Prévision optimale). La **prévision optimale** de X_{n+h} sachant $X_1, \dots, X_n$ au sens des moindres carrés est :

$$\hat{X}_{n+h|n} = \mathbb{E}[X_{n+h} | X_1, \dots, X_n].$$

L'erreur de prévision est $e_{n+h|n} = X_{n+h} - \hat{X}_{n+h|n}$.

Théorème 10.2 (Erreur quadratique moyenne). *L'espérance conditionnelle minimise l'EQM :*

$$\hat{X}_{n+h|n} = \arg \min_{g(X_1, \dots, X_n)} \mathbb{E} [(X_{n+h} - g)^2].$$

10.3 Prévision ARMA

Théorème 10.3 (Prévision d'un ARMA(p, q)). *En utilisant la représentation MA(∞) : $X_t = \sum_{j=0}^{\infty} \psi_j \varepsilon_{t-j}$, la prévision à h pas est :*

$$\hat{X}_{n+h|n} = \sum_{j=h}^{\infty} \psi_j \varepsilon_{n+h-j}.$$

L'erreur de prévision est $e_{n+h|n} = \sum_{j=0}^{h-1} \psi_j \varepsilon_{n+h-j}$, de variance :

$$\sigma_h^2 = \sigma^2 \sum_{j=0}^{h-1} \psi_j^2.$$

Corollaire 10.4. *Pour un ARMA stationnaire, $\sigma_h^2 \rightarrow \gamma(0) = \text{Var}(X_t)$ quand $h \rightarrow \infty$: la prévision converge vers la moyenne inconditionnelle et l'intervalle de confiance s'élargit vers l'intervalle inconditionnel.*

10.4 Intervalles de prévision

Définition 10.5. Sous l'hypothèse gaussienne, un intervalle de prévision à $(1 - \alpha)$ pour X_{n+h} est :

$$\hat{X}_{n+h|n} \pm z_{1-\alpha/2} \sigma_h,$$

où $z_{1-\alpha/2}$ est le quantile de la loi normale.

Attention

Les intervalles de prévision ne tiennent pas compte de l'incertitude sur les paramètres estimés. Des méthodes de bootstrap ou de simulation peuvent être utilisées pour des intervalles plus réalistes.

10.5 Prévision ARIMA

Théorème 10.6. Pour un $ARIMA(p, d, q)$, la prévision à long terme du niveau X_{n+h} croît linéairement (si $d = 1$) ou quadratiquement (si $d = 2$) : l'intervalle de prévision s'élargit sans borne.

10.6 Critères d'évaluation

Définition 10.7 (Métriques de prévision).

$$\text{MAE} = \frac{1}{H} \sum_{h=1}^H |e_{n+h|n}|, \quad (10.1)$$

$$\text{RMSE} = \sqrt{\frac{1}{H} \sum_{h=1}^H e_{n+h|n}^2}, \quad (10.2)$$

$$\text{MAPE} = \frac{100}{H} \sum_{h=1}^H \frac{|e_{n+h|n}|}{|X_{n+h}|} \%. \quad (10.3)$$

Définition 10.8 (Validation croisée temporelle). Pour évaluer la capacité prédictive, on utilise un découpage **train/test** respectant l'ordre temporel (pas de shuffling). La méthode **rolling window** ou **expanding window** est standard.

10.7 Test de Diebold–Mariano

Définition 10.9. Pour comparer deux modèles de prévision, soit $d_t = L(e_{1,t}) - L(e_{2,t})$ la différence de perte. Sous H_0 (égalité) :

$$\text{DM} = \frac{\bar{d}}{\sqrt{\hat{\sigma}_d^2/H}} \xrightarrow{d} \mathcal{N}(0, 1).$$

10.8 Implémentation

Python

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.arima.model import ARIMA
from sklearn.metrics import mean_absolute_error, mean_squared_error

# Simuler un ARMA(1,1)
np.random.seed(42)
from statsmodels.tsa.arima_process import ArmaProcess
ar = np.array([1, -0.7])
ma = np.array([1, 0.3])
data = ArmaProcess(ar, ma).generate_sample(nsample=300)

# Train/test split
train, test = data[:250], data[250:]

# Ajuster et prévoir
model = ARIMA(train, order=(1, 0, 1))
results = model.fit()

forecast = results.get_forecast(steps=50)
pred = forecast.predicted_mean
ci = forecast.conf_int(alpha=0.05)

# Métriques
mae = mean_absolute_error(test, pred)
rmse = np.sqrt(mean_squared_error(test, pred))
print(f"MAE = {mae:.4f}")
print(f"RMSE = {rmse:.4f}")

# Graphique
fig, ax = plt.subplots(figsize=(12, 5))
ax.plot(range(250), train, 'b-', lw=0.7, label='Train')
ax.plot(range(250, 300), test, 'g-', lw=1, label='Test')
ax.plot(range(250, 300), pred, 'r--', lw=1.5, label='Prévision')
ax.fill_between(range(250, 300), ci[:, 0], ci[:, 1],
                alpha=0.2, color='red')
ax.legend()
ax.set_title('Prévision ARMA(1,1)')
plt.savefig('ch10_forecast.pdf')
plt.show()

# Rolling forecast
rolling_preds = []
for i in range(len(test)):
    train_i = data[:250+i]
    model_i = ARIMA(train_i, order=(1,0,1)).fit()
```

```

rolling_preds.append(model_i.forecast(steps=1)[0])
rmse_roll = np.sqrt(mean_squared_error(test, rolling_preds))
print(f"RMSE rolling: {rmse_roll:.4f}")

```

10.9 Exercices

Exercice 10.1 (*). Pour un AR(1) avec $\phi = 0,8$ et $\sigma^2 = 1$, calculer $\hat{X}_{n+1|n}$, $\hat{X}_{n+2|n}$ et les variances de prévision correspondantes.

Exercice 10.2 (**). Montrer que la variance de prévision à h pas d'un AR(1) est $\sigma^2 \frac{1-\phi^{2h}}{1-\phi^2}$.

Exercice 10.3 (**). Expliquer pourquoi le MAPE est problématique quand X_t prend des valeurs proches de zéro.

Exercice 10.4 (*** – Projet). Comparer les prévisions hors échantillon d'un ARIMA, d'un lissage exponentiel et d'un modèle naïf ($\hat{X}_{n+h} = X_n$) sur 5 séries réelles. Utiliser le test de Diebold–Mariano.

Formules clés

$$\hat{X}_{n+h|n} = \sum_{j=h}^{\infty} \psi_j \varepsilon_{n+h-j}, \quad \sigma_h^2 = \sigma^2 \sum_{j=0}^{h-1} \psi_j^2$$

$$\text{IC}_{95\%} : \hat{X}_{n+h|n} \pm 1,96 \sigma_h$$

$$\text{RMSE} = \sqrt{\frac{1}{H} \sum e_{n+h}^2}$$

Chapitre 11

Applications : Finance, Économie, Météorologie

11.1 Exemple motivant

Les outils développés dans ce cours s'appliquent directement à des problèmes concrets : gestion de portefeuille, prévision macroéconomique, modélisation climatique.

11.2 Finance : Modélisation de la volatilité

11.2.1 Valeur à risque (VaR)

Définition 11.1 (VaR). La **Value at Risk** au niveau α est le quantile α de la distribution des pertes :

$$P(L_t > \text{VaR}_\alpha) = \alpha.$$

Avec un modèle GARCH : $\text{VaR}_\alpha = -\hat{\mu}_t + \hat{\sigma}_t q_\alpha$, où q_α est le quantile de la distribution des innovations.

11.2.2 Application

Python — VaR avec GARCH

```
import numpy as np
import pandas as pd
from arch import arch_model

# Simuler des rendements type financier
np.random.seed(42)
n = 2000
omega, alpha, beta = 0.01, 0.08, 0.90
sigma2 = np.zeros(n)
r = np.zeros(n)
sigma2[0] = omega / (1 - alpha - beta)
for t in range(1, n):
    sigma2[t] = omega + alpha * r[t-1]**2 + beta * sigma2[t-1]
    r[t] = np.sqrt(sigma2[t]) * np.random.standard_t(5)
```

```
# Estimer GARCH(1,1) avec innovations t de Student
am = arch_model(r * 100, vol='Garch', p=1, q=1, dist='t')
res = am.fit(disp='off')

# VaR 1%
forecasts = res.forecast(horizon=1)
sigma_next = np.sqrt(forecasts.variance.values[-1, 0])
from scipy.stats import t as t_dist
nu = res.params['nu']
VaR_01 = sigma_next * t_dist.ppf(0.01, nu)
print(f"VaR 1% = {VaR_01:.4f}%")
```

11.3 Économie : Prévision macroéconomique

11.3.1 PIB et indicateurs avancés

Méthode

1. Charger les séries trimestrielles (PIB, consommation, investissement).
2. Tester la stationnarité (ADF). Différencier si nécessaire.
3. Estimer un VAR ou un ARIMA saisonnier.
4. Évaluer par validation hors échantillon.

Python — Prévision macro

```
import pandas as pd
import numpy as np
from statsmodels.tsa.api import VAR
from statsmodels.tsa.stattools import adfuller

# Exemple avec données simulées (remplacer par FRED/Eurostat)
np.random.seed(0)
n = 200
gdp = np.cumsum(np.random.normal(0.5, 1, n))      # trend + bruit
cpi = 0.3 * gdp + np.cumsum(np.random.normal(0, 0.5, n))

df = pd.DataFrame({'GDP_growth': np.diff(gdp),
                   'Inflation': np.diff(cpi)})

# VAR
model = VAR(df)
results = model.fit(ic='bic')
print(f"Ordre sélectionné: VAR({results.k_ar})")
print(results.summary())
```

```
# Prédiction
forecast = results.forecast(df.values[-results.k_ar:], steps=8)
print("Prédiction PIB (8 trimestres):", forecast[:, 0])
```

11.4 Météorologie : Températures et saisonnalité

Python — Série de températures

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.seasonal import seasonal_decompose
from statsmodels.tsa.statespace.sarimax import SARIMAX

# Simuler des températures mensuelles
np.random.seed(42)
n_years = 30
t = np.arange(n_years * 12)
seasonal = 10 * np.sin(2 * np.pi * t / 12)
trend = 0.01 * t # réchauffement
noise = np.random.normal(0, 2, len(t))
temp = 15 + trend + seasonal + noise

df = pd.DataFrame({'temp': temp},
                  index=pd.date_range('1990-01', periods=len(t), freq='ME'))

# Décomposition
decomp = seasonal_decompose(df['temp'], model='additive', period=12)
decomp.plot()
plt.savefig('ch11_decomp.pdf')
plt.show()

# SARIMA
train = df.iloc[:-24]
test = df.iloc[-24:]
model = SARIMAX(train, order=(1,0,1),
                 seasonal_order=(1,1,1,12))
results = model.fit(dispatch=False)
pred = results.forecast(steps=24)

rmse = np.sqrt(np.mean((test['temp'] - pred.values)**2))
print(f"RMSE prévision 24 mois: {rmse:.2f}°C")
```

11.5 Résumé méthodologique

Domaine	Modèle typique	Enjeu principal
Finance	GARCH, EGARCH	Volatilité, queues lourdes
Macroéconomie	VAR, VECM, ARIMA	Causalité, coïntégration
Météorologie	SARIMA, espace d'états	Saisonnalité, tendance
Épidémiologie	SIR + espace d'états	Non-linéarité
Signal	Analyse spectrale	Périodicités cachées

11.6 Exercices

Exercice 11.1 (★). Calculer la VaR à 5% pour un GARCH(1,1) avec $\hat{\sigma}_{t+1} = 2\%$ et innovations gaussiennes.

Exercice 11.2 (★★). Comparer la prévision SARIMA et le lissage exponentiel de Holt–Winters sur les températures mensuelles. Lequel est le plus précis ?

Exercice 11.3 (★★★ – Projet). Choisir un jeu de données réel dans l'un des trois domaines. Appliquer la méthodologie complète : exploration, modélisation (au moins 3 modèles), diagnostic, prévision hors échantillon, comparaison. Rédiger un rapport de 10 pages.

Formules clés

$$\begin{aligned} \text{VaR}_\alpha &= -\hat{\mu}_t + \hat{\sigma}_t q_\alpha \\ \text{RMSE} &= \sqrt{\frac{1}{H} \sum (X_{n+h} - \hat{X}_{n+h})^2} \\ \text{DM} &= \frac{\bar{d}}{\sqrt{\hat{\sigma}_d^2/H}} \xrightarrow{d} \mathcal{N}(0, 1) \end{aligned}$$

Formulaire

.1 Processus ARMA

$$\text{AR}(p) : X_t = \phi_1 X_{t-1} + \cdots + \phi_p X_{t-p} + \varepsilon_t \quad (1)$$

$$\text{MA}(q) : X_t = \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} \quad (2)$$

$$\text{ARMA}(p, q) : \phi(B)X_t = \theta(B)\varepsilon_t \quad (3)$$

$$\text{ARIMA}(p, d, q) : \phi(B)(1 - B)^d X_t = \theta(B)\varepsilon_t \quad (4)$$

.2 Autocovariance et autocorrélation

$$\gamma(h) = \text{Cov}(X_t, X_{t+h}) \quad (5)$$

$$\rho(h) = \gamma(h)/\gamma(0) \quad (6)$$

$$\hat{\gamma}(h) = \frac{1}{n} \sum_{t=1}^{n-h} (X_t - \bar{X})(X_{t+h} - \bar{X}) \quad (7)$$

.3 Critères d'information

$$\text{AIC} = -2 \ln L + 2k \quad (8)$$

$$\text{BIC} = -2 \ln L + k \ln n \quad (9)$$

.4 GARCH

$$\text{GARCH}(1, 1) : \sigma_t^2 = \omega + \alpha \varepsilon_{t-1}^2 + \beta \sigma_{t-1}^2 \quad (10)$$

.5 Filtre de Kalman

$$\text{État} : \mathbf{x}_{t+1} = F\mathbf{x}_t + \mathbf{w}_t \quad (11)$$

$$\text{Observation} : \mathbf{y}_t = H\mathbf{x}_t + \mathbf{v}_t \quad (12)$$

.6 VAR

$$\text{VAR}(p) : \mathbf{Y}_t = \mathbf{c} + A_1 \mathbf{Y}_{t-1} + \cdots + A_p \mathbf{Y}_{t-p} + \mathbf{u}_t \quad (13)$$

Bibliographie

- [1] BOX, G.E.P., Jenkins, G.M., Reinsel, G.C. & Ljung, G.M. (2015). *Time Series Analysis : Forecasting and Control*. 5e éd., Wiley.
- [2] BROCKWELL, P.J. & Davis, R.A. (2016). *Introduction to Time Series and Forecasting*. 3e éd., Springer.
- [3] HAMILTON, J.D. (1994). *Time Series Analysis*. Princeton University Press.
- [4] SHUMWAY, R.H. & Stoffer, D.S. (2017). *Time Series Analysis and Its Applications*. 4e éd., Springer.
- [5] GOURIÉROUX, C. & Monfort, A. (1995). *Séries Temporelles et Modèles Dynamiques*. 2e éd., Economica.
- [6] TSAY, R.S. (2010). *Analysis of Financial Time Series*. 3e éd., Wiley.