

Econométrie des séries temporelles multivariées

Théorie et application sous Python

Duvérier DJIFACK ZEBAZE



Econométrie des séries temporelles multivariées

Théorie et application sous Python

Duvérier DJIFACK ZEBAZE

Table des matières

I	Économétrie des processus stationnaires	1
1	Introduction aux processus VAR	2
1.1	Notations et quelques définitions	2
1.1.1	Généralités des processus d —vectoriel	2
1.1.2	Quelques outils mathématiques	5
1.2	Processus vectoriel autorégressif	9
1.2.1	Processus VAR(p)	9
1.2.2	Fonction d'autocovariance et d'autocorrélation	17
1.2.3	Fonction d'autocorrélation partielle et densité spectrale	26
2	Estimation et validation d'un processus VAR	27
2.1	Estimation des paramètres d'un VAR(p)	27
2.1.1	Estimation par moindres carrés généralisés	27
2.1.2	Estimation des moindres carrés avec données ajustées à la moyenne	35
2.1.3	Estimation par maximum de vraisemblance	38
2.1.4	Estimation par équation de Yule - Walker	38
2.2	Détermination du nombre de retards	40
2.2.1	Test du rapport de maximum de vraisemblance	40
2.2.2	Les critères d'information	41
2.3	Validation de modèle et prévision	43
2.3.1	Tests de bruit blanc des erreurs	44
2.3.2	Test d'hétéroscédasticité	48
2.3.3	Test de normalité	48
2.3.4	Prévision	53

3	Analyse structurelle d'un processus VAR	60
3.1	Concept de causalité	60
3.1.1	Causalité au sens de Pierre et Haugh	61
3.1.2	Causalité au sens de Granger	61
3.1.3	Causalité au sens de Sims	64
3.1.4	Mesure et test de causalité	65
3.1.5	Exogénéité	73
3.2	Analyse de réponse impulsionnelle	73
3.2.1	Fonctions de réponse simples	74
3.2.2	Fonctions de réponse orthogonalisées	78
3.2.3	Interprétation des chocs orthogonalisés et conséquence de cette orthogonalisation	80
3.2.4	Intervalle de confiance des fonctions de réponses	83
3.2.5	Décomposition de la variance des erreurs de prévisions	85
II	Économétrie des processus non stationnaires	88
4	Cointégration et modèle à correction d'erreur	89
4.1	La notion de cointégration	89
4.1.1	Rappels sur les tests de racine unitaire	89
4.1.2	Justification de la cointégration	89
4.1.3	Définition de la cointégration	90
4.2	Le modèle à correction d'erreur	91
4.2.1	Estimation du modèle à correction d'erreur	91
4.2.2	Approche multivariée de la cointégration : l'analyse de Johansen	103
5	Validation des modèles VECM	120
5.1	Diagnostic des résidus et prévision	120
5.1.1	Test d'autocorrélation des résidus	120
5.1.2	Test de normalité	122
5.1.3	Prévision	123
5.2	Analyse structurelle d'un modèle VECM	126
5.2.1	Causalité au sens de Granger	126
5.2.2	Causalité de Granger au sens de Toda - Yamamoto	130
5.2.3	Causalité instantanée	131
5.2.4	Analyse des réponses impulsionnelles	133
6	Modélisation ARDL et test de cointégration aux bornes	135

6.1	Modélisation ARDL	135
6.1.1	Présentation générale	135
6.1.2	Estimation des paramètres d'un modèle ARDL	137
6.1.3	Décalage optimal	138
6.1.4	Prévision avec un modèle ARDL	139
6.2	Test de cointégration de Pesaran et <i>al.</i> (2001)	141
6.2.1	Rappel sur la cointégration	141
6.2.2	Le modèle ARDL Pesaran, Shin and Smith (2001)	142
	Tables statistiques	155
	Bibliographie	157

Première partie

Économétrie des processus stationnaires

Introduction aux processus VAR

Sommaire

1.1 Notations et quelques définitions	2
1.2 Processus vectoriel autorégressif	9

Les processus VAR(p) (Vector AutoRegressive) constituent une généralisation des processus AR au cas multivarié. Ils ont été introduits par Sims (1980) comme alternative aux modèles macroéconométriques d'inspiration keynésienne. Selon Sims (1980), ces modèles macroéconométriques souffrent d'un certain nombre d'insuffisances, tels que : les restrictions a priori trop fortes sur les paramètres par rapport à ce que prédit la théorie ; l'absence de tests sérieux sur la structure causale et le traitement inadéquat des anticipations. En outre, d'un point de vue empirique, ces modèles ont été mis à mal par les événements survenus au cours des années 70 (chocs pétroliers, récession mondiale, etc.) conduisant à de très importantes erreurs de prévision. Pour ces différentes raisons, Sims (1980) a proposé une modélisation multivariée sans autre restriction a priori que le choix des variables sélectionnées et du nombre des retards. La modélisation VAR repose sur l'hypothèse selon laquelle l'évolution de l'économie est bien approchée par la description du comportement dynamique d'un vecteur de d variables dépendant linéairement du passé. Depuis les travaux initiaux de Sims (1980), les techniques économétriques basées sur les modèles VAR ont connu de nombreux développements.

1.1 Notations et quelques définitions

1.1.1 Généralités des processus d -vectoriel

On considère un processus $(X_t)_{t \in \mathbb{Z}}$ à valeurs dans \mathbb{R}^d . Par convention, X_t est un vecteur colonne de variables aléatoires notées $X_{1,t}, X_{2,t}, \dots, X_{d,t}$

$$X_t = (X_{1,t}, \dots, X_{d,t})' \quad (1.1)$$

Chacune des composantes $X_{j,t}$, $j = 1, \dots, d$ $t \in \mathbb{Z}$, est une variable aléatoire réelle. La loi de $(X_t)_{t \in \mathbb{Z}}$ est souvent résumée par ses moments d'ordre 1 et 2 :

1. La moyenne du processus X_t donnée par :

$$\mathbb{E}[X_t] = \begin{pmatrix} \mathbb{E}(X_{1t}) \\ \vdots \\ \mathbb{E}(X_{dt}) \end{pmatrix} = \begin{pmatrix} \mu_{1t} \\ \vdots \\ \mu_{dt} \end{pmatrix} = \mu_t \quad (1.2)$$

où μ_t est un vecteur $d \times 1$. Bien entendu, lorsque $\mu_t = 0$, le processus X_t est dit centré.

2. La matrice d'autocovariance, noté $\Gamma(t, h)$ et donnée par :

$$\Gamma(t, h) = \text{Cov}(X_t, X_{t-h}) = \mathbb{E} \left[(X_t - \mu_t)(X_{t-h} - \mu_{t-h})' \right] = \Sigma_{X_t, X_{t-h}} \quad (1.3)$$

Notons $\Sigma_{X_t, X_{t-h}}(i, j) = \gamma_{i,j}(t, h)$ le terme général de cette matrice, qui représente pour $i \neq j$, la fonction de covariance croisée de deux séries $X_{i,t}$ et $X_{j,t}$:

$$\gamma_{i,j}(t, h) = \begin{pmatrix} \gamma_{1,1}(t, h) & \cdots & \gamma_{1,d}(t, h) \\ & \ddots & \vdots \\ & & \gamma_{d,d}(t, h) \end{pmatrix} \quad (1.4)$$

1.1.1.1 Stationnarité d'un processus d -vectoriel

On généralise la notion de stationnarité faible ou de second ordre vue dans les chapitres précédents au cas vectoriel.

Définition 1.1 *Stationnarité faible d'un processus d -vectoriel*

Le processus d -vectoriel $(X_t)_{t \in \mathbb{Z}}$ est dit stationnaire au sens faible s'il existe $\mu \in \mathbb{R}^k$ et une séquence $(\Gamma_X(h))_{h \geq 0}$ de matrice de format $d \times d$ tel que :

- $\forall t \in \mathbb{Z}, \quad \mathbb{E}(X_t) = \mu$
- $\forall t \in \mathbb{Z}, \quad \mathbb{E} \left[(X_t - \mu)(X_t - \mu)' \right] < \infty$
- $\forall t \in \mathbb{Z}, \forall h \in \mathbb{N}, \quad \mathbb{E} \left[(X_t - \mu)(X_{t-h} - \mu)' \right] = \Gamma_X(h)$

Remarque 1.1

- $\Gamma_X(\cdot)$ est la fonction matrice de covariance du processus stationnaire X_t .
- La stationnarité du processus d -dimensionnel implique la stationnarité de chacun des composantes du processus d -vectoriel.
- La stationnarité au sens strict est similaire à celle d'un processus univarié.

Proposition 1.1

La matrice d'autocovariance $\Gamma(h)$ d'un processus X_t stationnaire satisfait les propriétés suivantes :

- $\Gamma_X(-h) = \Gamma_X'(h) \forall h = 0, 1, \dots$
- $|\gamma_{i,j}(h)| \leq |\gamma_{i,i}(0)\gamma_{j,j}(0)|^{1/2}, \forall h = 0, 1, \dots, \forall i \text{ et } \forall j = 1, \dots, d.$
- $\gamma_{i,i}(\cdot)$ est la fonction d'autocovariance de la composante i $i = 1, \dots, d.$

La stationnarité du processus d -vectoriel implique la stationnarité de chacun de ses composantes. En effet, le vecteur μ est composé de valeurs constantes qui représentent les moyennes de chaque composante du processus. Par ailleurs, la fonction d'autocovariance d'un processus d -vectoriel stationnaire à l'horizon h dépend uniquement de l'écart de temps h . Ceci entraîne que chacun des éléments de la matrice d'autocovariance dépend uniquement de h . Par contre, la stationnarité marginale des d composantes n'entraîne pas leur stationnarité conjointe. En voici un exemple illustratif. :

Exemple 1.1

Supposons deux processus stationnaires $X_{1,t}$ et $X_{2,t}$ et un processus bivarié X_t défini comme suit :

$$X_t = \begin{pmatrix} X_{1,t} \\ X_{2,t} \end{pmatrix} = \begin{pmatrix} \varepsilon_t \\ \eta_t + (-1)^t \varepsilon_t \end{pmatrix} \quad \forall t \in \mathbb{Z} \quad (1.5)$$

où ε_t et η_t sont deux bruits blancs faibles indépendants. On montre aisément que

$$\begin{cases} \mathbb{E}(X_{1,t}) &= 0 \\ \text{Cov}(X_{1,t}, X_{1,t+h}) &= \mathbb{E}(X_{1,t}X_{1,t+h}) = \mathbb{E}(\varepsilon_t \varepsilon_{1,t+h}) = \gamma_\varepsilon(h) \end{cases} \quad (1.6)$$

et

$$\begin{cases} \mathbb{E}(X_{2,t}) &= 0 \\ \text{Cov}(X_{2,t}, X_{2,t+h}) &= \mathbb{E}(X_{2,t}X_{2,t+h}) = \gamma_\eta(h) + (-1)^h \gamma_\varepsilon(h) \end{cases} \quad (1.7)$$

car ε_t et η_t sont indépendants. Par conséquent, $X_{1,t}$ et $X_{2,t}$ sont stationnaires.

Cependant, le processus bivarié X_t est-il stationnaire ? En effet, la matrice d'autocovariance associé à X_t s'écrit :

$$\begin{aligned} \mathbb{E}[(X_t - \mu)(X_{t+h} - \mu)'] &= \mathbb{E}(X_t X_{t+h}') = \mathbb{E} \left[\begin{pmatrix} X_{1,t} \\ X_{2,t} \end{pmatrix} (X_{1,t+h} \quad X_{2,t+h}) \right] \\ &= \mathbb{E} \begin{bmatrix} X_{1,t}X_{1,t+h} & X_{1,t}X_{2,t+h} \\ X_{2,t}X_{1,t+h} & X_{2,t}X_{2,t+h} \end{bmatrix} = \begin{bmatrix} \gamma_\varepsilon(h) & (-1)^{t+h} \gamma_\varepsilon(h) \\ (-1)^t \gamma_\varepsilon(h) & \gamma_\eta(h) + (-1)^h \gamma_\varepsilon(h) \end{bmatrix} \end{aligned}$$

Comme $\mathbb{E}[(X_t - \mu)(X_{t+h} - \mu)']$ dépend de t , alors le processus bivarié X_t n'est pas stationnaire.

1.1.1.2 Bruit blanc multivarié et moyenne mobile d'ordre infini**Définition 1.2** *Bruit blanc multivarié*

Un bruit blanc faible multivarié $(\varepsilon_t)_{t \in \mathbb{Z}}$ de variance Σ_ε satisfait :

- $\mathbb{E}(\varepsilon_t) = 0, \forall t \in \mathbb{Z}$
- $\mathbb{E} \begin{bmatrix} \varepsilon_t \varepsilon_s' \end{bmatrix} = 0, \forall s \neq t \in \mathbb{Z}.$
- $\mathbb{E} \begin{bmatrix} \varepsilon_t \varepsilon_t' \end{bmatrix} = \Sigma_\varepsilon \quad \forall t \in \mathbb{Z}.$

où Σ_ε est la matrice de variance covariance constante des d composantes du processus ε_t .

Définition 1.3 *Processus vectoriel moyenne mobile d'ordre infini*

Un processus vectoriel moyenne mobile (VMA, *Vector Moving Average*) d'ordre infini est défini par :

$$X_t = \mu + \sum_{j \geq 0} \Psi_j \varepsilon_{t-j} \quad (1.8)$$

où $(\varepsilon_t)_{t \in \mathbb{Z}}$ est un bruit blanc faible. On suppose que $\sum_{j \geq 0} |\psi_{i,l}^j| < +\infty, \forall 1 \leq i, l \leq d$

1.1.1.3 Autocovariance et autocorrélation d'ordre h

Définition 1.4 *Autocovariance d'ordre h*

La matrice d'autocovariance d'ordre h du processus X est donnée par :

$$\Gamma_X(h) = \text{Cov}(X_t, X_{t-h}) = \text{Cov}\left(\mu + \sum_{j \geq 0} \Psi_j \varepsilon_{t-j}, \mu + \sum_{k \geq 0} \Psi_k \varepsilon_{t-k}\right) = \sum_{k \geq 0} \Psi_{h+k} \Sigma_\varepsilon \Psi_k' \quad (1.9)$$

Définition 1.5 *Autocorrélation d'ordre h*

La matrice d'autocorrélation d'ordre $h \geq 0$ est donnée par :

$$R_X(h) = D^{-1} \Gamma_X(h) D^{-1} \quad (1.10)$$

où $D = \text{diag} \left\{ \sqrt{V(X_{1,t})}, \dots, \sqrt{V(X_{d,t})} \right\}$.

1.1.2 Quelques outils mathématiques

1.1.2.1 Produit de Kronecker

Le produit de kronecker de la matrice A de format $(p \times q)$ avec la matrice B de format $(r \times s)$ est la matrice C de format $(pr \times qs)$ donnée par :

$$A \otimes B = \begin{pmatrix} a_{11}B & \dots & a_{1q}B \\ \dots & \dots & \dots \\ a_{p1}B & \dots & a_{pq}B \end{pmatrix} \quad (1.11)$$

En d'autres termes, $A \otimes B$ signifie qu'on effectue le produit de tous les éléments de A par la matrice B .

Exemple 1.2

Soient A et B deux matrices définies par : $A = \begin{pmatrix} 3 & 1 & 0 \\ -1 & 4 & 2 \\ 0 & 1 & 0 \end{pmatrix}$ et $B = \begin{pmatrix} 1 & 7 \\ 2 & -1 \end{pmatrix}$

Alors, le produit de kronecker $A \otimes B$ est donné par :

$$A \otimes B = \begin{pmatrix} 3 \begin{pmatrix} 1 & 7 \\ 2 & -1 \end{pmatrix} & 1 \begin{pmatrix} 1 & 7 \\ 2 & -1 \end{pmatrix} & 0 \\ -1 \begin{pmatrix} 1 & 7 \\ 2 & -1 \end{pmatrix} & 4 \begin{pmatrix} 1 & 7 \\ 2 & -1 \end{pmatrix} & 2 \begin{pmatrix} 1 & 7 \\ 2 & -1 \end{pmatrix} \\ 0 & 1 \begin{pmatrix} 1 & 7 \\ 2 & -1 \end{pmatrix} & 0 \end{pmatrix} = \begin{pmatrix} 3 & 21 & 1 & 7 & 0 & 0 \\ 6 & -3 & 2 & -1 & 0 & 0 \\ -1 & -7 & 4 & 28 & 2 & 14 \\ -2 & 1 & 8 & -4 & 4 & -2 \\ 0 & 0 & 1 & 7 & 0 & 0 \\ 0 & 0 & 2 & -1 & 0 & 0 \end{pmatrix} \quad (1.12)$$

Sous Python, la fonction `np.kron` permet d'effectuer le produit de kronecker entre deux matrices. Vérifions nos résultats avec cette fonction.

```
# produit de kronecker
import numpy as np
A = np.array([[3,1,0],[-1,4,3],[0,1,0]])
B = np.array([[1,7],[2,-1]])
kron = np.kron(A,B)
print(kron)

## [[ 3 21  1  7  0  0]
##   [ 6 -3  2 -1  0  0]
##   [-1 -7  4 28  3 21]
##   [-2  1  8 -4  6 -3]
##   [ 0  0  1  7  0  0]
##   [ 0  0  2 -1  0  0]]
```

On constate que le résultat est le même.

Propriété 1.1 Propriétés de \otimes

L'opérateur \otimes du produit de kronecker vérifie les propriétés suivantes :

- $A \otimes B \neq B \otimes A$
- $A \otimes (B + \lambda C) = A \otimes B + \lambda(A \otimes C)$
- $(A \otimes B)(C \otimes D) = AC \otimes BD$
- Si A et B sont inversibles, alors :

$$(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$$

- Si A est de format $(m \times m)$ et B de format $(n \times n)$, alors :

$$\det(A \otimes B) = (\det(A))^n \times (\det(B))^m$$

- Si A et B sont deux matrices carrées, alors :

$$\text{tr}(A \otimes B) = \text{tr}(A) \times \text{tr}(B)$$

1.1.2.2 Opérateurs vec et vech

Définition 1.6 Opérateur vec

Soit $A = (a_1, \dots, a_n)$ une matrice de format $(m \times n)$ avec $(m \times 1)$ colonnes a_i . L'opérateur vec (*vectorization*) transforme la matrice A en un vecteur de format $(mn \times 1)$:

$$\text{vec}(A) = \begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix}, \quad \text{avec} \quad a_i = \begin{pmatrix} a_{1i} \\ \vdots \\ a_{mi} \end{pmatrix}$$

Exemple 1.3

Soient A et B deux matrices définies par :

$$A = \begin{pmatrix} 3 & 1 & 0 \\ -1 & 4 & 2 \\ 0 & 1 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} 1 & 7 \\ 2 & -1 \end{pmatrix}$$

On obtient :

$$\text{vec}(A) = \begin{pmatrix} 3 \\ -1 \\ 0 \\ 1 \\ 4 \\ 1 \\ 0 \\ 2 \\ 0 \end{pmatrix} \quad \text{et} \quad \text{vec}(B) = \begin{pmatrix} 1 \\ 2 \\ 7 \\ -1 \end{pmatrix} \quad (1.13)$$

Il n'existe pas de fonction en python adéquat pour faire cette opération, mais toutefois, pour une matrice A de format $(n \times m)$, on peut utiliser la fonction `np.reshape` qui nous retournera un vecteur de format $(nm \times 1)$. Nous allons créer une fonction `vec`.

```
# Opérateur vec
def vec(A):
    return A.reshape((-1, 1), order="F")
vecA = vec(A)
print(vecA)

## [[ 3]
##  [-1]
##   [ 0]
##   [ 1]
##   [ 4]
##   [ 1]
##   [ 0]
##   [ 3]
##   [ 0]]
```

Propriété 1.2 Propriétés de `vec`

L'opérateur `vec` vérifie les propriétés suivantes :

1. L'opérateur `vec` est linéaire : $\text{vec}(A + B) = \text{vec}(A) + \text{vec}(B)$
- 2.

$$\begin{aligned} \text{vec}(AB) &= \text{vec}(ABI) = \text{vec}(IAB) \\ &= (I \otimes A)\text{vec}(B) \\ &= (B' \otimes I)\text{vec}(A) \end{aligned}$$

3.

$$\begin{aligned}
\text{vec}(ABC) &= \text{vec}[IA(BC)] = \text{vec}[(AB)CI] \\
&= (C' \otimes A)\text{vec}(B) \\
&= (I \otimes AB)\text{vec}(C) \\
&= (C'B' \otimes I)\text{vec}(A)
\end{aligned}$$

$$4. \text{vec}(B')'\text{vec}(A) = \text{tr}(BA) = \text{tr}(AB) = \text{vec}(A')'\text{vec}(B)$$

5.

$$\begin{aligned}
\text{tr}(ABC) &= \text{vec}(A')'(C' \otimes I)\text{vec}(B) \\
&= \text{vec}(A')'(I \otimes B)\text{vec}(C) \\
&= \text{vec}(B')'(A' \otimes I)\text{vec}(C) \\
&= \text{vec}(B')'(I \otimes C)\text{vec}(A) \\
&= \text{vec}(C')'(B' \otimes I)\text{vec}(A) \\
&= \text{vec}(C')'(I \otimes A)\text{vec}(B)
\end{aligned}$$

Définition 1.7 *Opérateur vech*

Soit A une matrice carrée d'ordre n . L'opérateur vech (*half-vectorization*) transforme les éléments sous la diagonale principale de A en un vecteur. Par exemple :

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \implies \text{vech}(A) = \begin{pmatrix} a_{11} \\ a_{21} \\ a_{31} \\ a_{22} \\ a_{32} \\ a_{33} \end{pmatrix}$$

Plus généralement, si A est une matrice de format n , alors $\text{vech}(A)$ est un vecteur de dimension $n(n+1)/2$. L'opérateur vech est généralement appliqué aux matrices symétriques pour collecter uniquement les éléments séparés.

Exemple 1.4

Soit A la matrice symétrique suivante définie par :

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 5 & 6 \\ 3 & 6 & 9 \end{pmatrix} \implies \text{vech}(A) = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 5 \\ 6 \\ 9 \end{pmatrix}$$

Tout comme pour la fonction vec , il n'existe pas de fonction en python adéquat pour faire cette opération. Nous allons créer une fonction [vech](#).

```

# Opérateur vech
def vech(x):
    A = np.tril(x).reshape((-1,1),order="F")
    return A[A!=0].reshape((-1,1),order="F")
# Application
A = np.array([[1,2,3],[2,5,6],[3,6,9]])
print(vech(A))

## [[1]
##  [2]
##  [3]
##  [5]
##  [6]
##  [9]]

```

1.2 Processus vectoriel autorégressif

1.2.1 Processus VAR(p)

Définition 1.8 *Processus VAR(p)*

Soit $(\varepsilon_t)_{t \in \mathbb{Z}}$ un bruit blanc faible multivarié de matrice de covariance Σ_ε non singulière. Un processus multivarié $(X_t)_{t \in \mathbb{Z}}$ est un VAR(p) s'il existe $c \in \mathbb{R}^d$ et p matrices Φ_1, \dots, Φ_p de format $d \times d$ tel que :

$$X_t = c + \sum_{j=1}^p \Phi_j X_{t-j} + \varepsilon_t \quad (1.14)$$

En utilisant l'opérateur retard B , on a :

$$\Phi(B)X_t = c + \varepsilon_t \quad (1.15)$$

avec $\Phi(B) = I - \Phi_1 B - \Phi_2 B^2 + \dots + \Phi_p B^p$.

1.2.1.1 Processus VAR(1)

Considérons le processus VAR(1) défini par :

$$X_t = c + \Phi_1 X_{t-1} + \varepsilon_t \quad (1.16)$$

Supposons $X_0 \neq 0$. On montre aisément que

$$X_t = (I_d + \Phi_1 + \Phi_1^2 + \dots + \Phi_1^{t-1})c + \Phi_1^t X_0 + \sum_{j=0}^{t-1} \Phi_1^j \varepsilon_{t-j}$$

Par conséquent, les vecteurs $X_{1,t}, \dots, X_{d,t}$ sont uniquement déterminés par $X_0, \varepsilon_1, \dots, \varepsilon_t$.

Proposition 1.2 *Représentation de Wold d'un processus VAR(1)*

Si les valeurs propres de Φ_1 sont de module strictement inférieurs à 1, alors la suite $\Phi_1^j, j = 0, 1, \dots$ est absolument sommable. La somme $\sum_{j=0}^{t-1} \Phi_1^j \varepsilon_{t-j}$ existe en moyenne quadratique. On trouve

$$\begin{aligned} X_t &= (I_d - \Phi_1)^{-1} c + \sum_{j=0}^{t-1} \Phi_1^j \varepsilon_{t-j} \quad \forall t \in \mathbb{Z} \\ &= \mu + \sum_{j=0}^{\infty} \Phi_1^j \varepsilon_{t-j} \quad (\text{avec } \mu = (I_d - \Phi_1)^{-1} c) \quad \forall t \in \mathbb{Z} \end{aligned}$$

Remarque 1.2

On montre aisément que :

$$\begin{aligned} - \quad & \forall t \in \mathbb{Z}, \quad \mathbb{E}(X_t) = \mu. \\ - \quad & \forall h \in \mathbb{Z}, \quad \Gamma_X(h) = \text{Cov}(X_t, X_{t-h}) = \sum_{i=0}^{\infty} \Phi_1^{h+i} \Sigma_\varepsilon (\Phi_1^i)' \end{aligned}$$

Propriété 1.3 Condition de stabilité d'un VAR(1)

Si les valeurs propres de la matrice Φ_1 sont de module strictement inférieur à 1, alors le processus VAR(1) est dit stable. Cette condition est équivalente à :

$$\det(I_d - \Phi_1 z) \neq 0 \quad \text{pour } |z| \leq 1 \quad (1.17)$$

Propriété 1.4 Condition de stabilité d'un VAR(1) : propriété réciproque

Un processus VAR(1) est dit stable si le polynôme $\det(I_d - \Phi_1 z)$ n'a pas de racine sur le cercle d'unité, c'est-à-dire

$$\det(I_d - \Phi_1 z) \neq 0 \quad \text{pour } |z| \leq 1 \quad (1.18)$$

Exemple 1.5

Pour illustrer la stabilité, considérons le processus VAR(1) bivarié suivant :

$$X_t = \begin{pmatrix} 1 \\ 2 \end{pmatrix} + \begin{pmatrix} 0.7 & 0.1 \\ 0.6 & 0.25 \end{pmatrix} X_{t-1} + \varepsilon_t \quad (1.19)$$

Pour ce processus, le polynôme caractéristique est :

$$\det(I_2 - \Phi_1 z) = \det \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 0.7 & 0.1 \\ 0.6 & 0.25 \end{bmatrix} z \right) = \det \begin{bmatrix} 1 - 0.7z & -0.1z \\ -0.6z & 1 - 0.25z \end{bmatrix} = 0.235z^2 - 0.95z + 1$$

La fonction `np.roots` de la librairie Numpy permet de trouver les racines d'un polynôme de degré n . Nous l'utilisons afin d'obtenir les racines de notre polynôme de degré 2.

```
# Racine d'un polynôme
coeff = np.array([0.235,-0.95,1])
z = np.roots(coeff)
print(z.round(4))
```

```
## [2.0213+0.412j 2.0213-0.412j]
```

Les racines de notre polynôme sont complexes.

```
# Cercle unité
def plotroot(root,title= "AR roots",color="black",ax=None):
    from matplotlib.patches import Ellipse
    real = np.real([1/x for x in root])
    im = np.imag([1/x for x in root])
    # Représentation graphique
    if ax is None:
        ax = plt.gca()
    ax.scatter(real,im,color=color);
    ellipse=Ellipse((0,0),width=2,height=2,facecolor="none",edgecolor = "black");
    ax.add_patch(ellipse);
    ax.set_title(title);
    ax.set_xlabel("Re(1/root)");
    ax.set_ylabel("Im(1/root)");
    ax.axvline(x=0,linewidth=2, color='black');
    ax.axhline(y=0,linewidth=2, color='black');

# Représentation graphique
fig, axe = plt.subplots(figsize=(10,10))
plotroot(z,ax=axe, title = "VAR(1) roots");
plt.show()
```

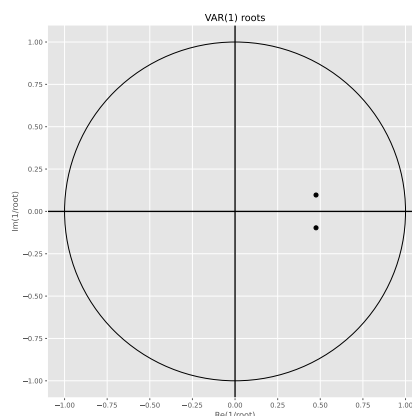


Figure 1.1 – Racines caractéristiques inverses du VAR(1)

On calcule les modules associés :


```
# Module
print(abs(z))
```

```
## [2.06284249 2.06284249]
```

Les modules associés sont supérieurs à 1, par conséquent le processus VAR(1) est stable.

Une deuxième approche pour vérifier la stabilité d'un processus VAR(1) est de calculer les valeurs propres associées à la matrice Φ_1 . Si ces valeurs propres sont de module inférieur à 1, alors le processus VAR(1) est stable. C'est cette seconde approche que nous utiliserons tout au long de ce chapitre pour vérifier la stabilité d'un processus VAR.

```
# Coefficient du VAR(1)
Phi1 = np.array([[0.7,0.1],[0.6,0.25]])
c = np.array([[1],[2]])
# Valeurs et vecteurs propres de Phi1
eigval, eigvect = np.linalg.eig(Phi1)
print(eigval)
```

```
## [0.80760337 0.14239663]
```

Les modules des valeurs propres de la matrice Φ_1 sont 0.8076 et 0.1424. Ces valeurs propres sont inférieures à 1, alors le processus VAR(1) est stable. On peut écrire le processus sous la forme moyenne mobile infinie :

$$X_t = (I_2 - \Phi_1)^{-1} c + \sum_{i \geq 0} \begin{pmatrix} 0.7 & 0.1 \\ 0.6 & 0.25 \end{pmatrix}^i \varepsilon_{t-i} \quad (1.20)$$

On définit une fonction `inv_ident_minus_matrix(.)` qui calcule $(I_2 - \Phi_1)^{-1}$ et on l'applique à notre matrice A.

```
# Calcul de l'inverse de I - Phi1
def inv_ident_minus_matrix(A):
    I = np.identity(A.shape[0])
    inv = np.linalg.inv(np.subtract(I,A))
    return inv
# Application
inv = inv_ident_minus_matrix(Phi1)
print(inv)
```

```
## [[4.54545455 0.60606061]
##  [3.63636364 1.81818182]]
```

On définit ensuite une seconde fonction `matrix_power_n(.,.)` qui permet de calculer la puissance n -ième d'une matrice A.

```
# Puissance n-ieme de Phi1
def matrix_power_n(A,n):
    B = A
```

```

for i in range(1,n):
    B = np.dot(A,B)
return B

```

Par exemple, on a Φ_1^2

```

# Calcul de Phi1^2
Phi1_square = matrix_power_n(Phi1,2) # Phi1 au carré
print(Phi1_square)

## [[0.55    0.095 ]
##  [0.57    0.1225]]

```

et Φ_1^3

```

# Calcul de Phi1^3
Phi1_cube = matrix_power_n(Phi1,3) # Phi1 au cube
print(Phi1_cube)

## [[0.442    0.07875 ]
##  [0.4725    0.087625]]

```

Exemple 1.6 Simulation d'un processus VAR(1) stationnaire

On considère le processus VAR(1) défini par l'expression (1.19) en supposant $\Sigma_\varepsilon = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}$.

On simule ce processus avec 250 observations.

```

# Simulation d'un VAR(1) bivarié
class VAR:
    def __init__(self, constant= np.array([0.0, 0.0]),
                 ar = np.array([[0.0,0.0],[0.0,0.0]]),
                 mean = np.array([0.0,0.0]),
                 cov = np.array([[0.0,0.0],[0.0,0.0]])):
        self.c = constant
        self.ar = ar
        self.mean = mean
        self.cov = cov
    def simulate(self, T=100):
        epsilon = np.random.multivariate_normal(self.mean,self.cov,T)
        X = np.zeros(shape=(T,2))
        X[0,:] = np.random.multivariate_normal((0,0),[[1, 0], [0, 1]],1)
        for t in range(1,T):
            X[t,0] = (self.c[0] + self.ar[0,0]*X[t-1,0]+self.ar[0,1]*X[t-1,1]+
                     epsilon[t,0])
            X[t,1] = (self.c[1] + self.ar[1,0]*X[t-1,0]+self.ar[1,1]*X[t-1,1]+
                     epsilon[t,1])
        return X

```

```
# Application
np.random.seed(12345)
Phi1 = np.array([[0.7,0.1],[0.6,0.25]])
c = np.array([[1],[2]])
covmat = np.array([[1.0,0.5],[0.5,1.0]])
T = 250
var1_sim = VAR(constant=c,ar=Phi1,cov=covmat).simulate(T=T)
print(var1_sim.shape)

## (250, 2)
```

On voit que notre programme simule deux processus de taille 250 chacun. On représente graphiquement nos séries.

```
# Représentation graphique
import matplotlib.pyplot as plt
fig, axe = plt.subplots(figsize=(16,6))
axe.plot(var1_sim[:,0],color='black',label = '$X_{1,t}$');
axe.plot(var1_sim[:,1],color='red',label = '$X_{2,t}$');
axe.set_xlabel('t');
axe.set_ylabel('$X_t$');
axe.legend();
plt.show()
```

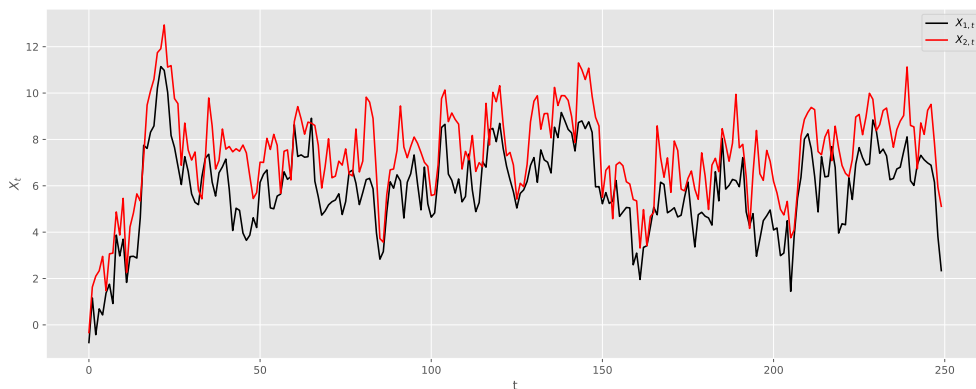


Figure 1.2 – Simulation d'un processus $VAR(1)$ bivarié stationnaire

1.2.1.2 Processus $VAR(p)$

Proposition 1.3 *Forme $VAR(1)$ d'un processus $VAR(p)$*

Un processus $VAR(p)$ s'écrit :

$$X_t = c + \Phi_1 X_{t-1} + \dots + \Phi_p X_{t-p} + \varepsilon_t, \quad \forall t \in \mathbb{Z} \quad (1.21)$$

Ce processus peut s'écrire sous la forme d'un $VAR(1)$ à dp -dimension

$$Y_t = \tilde{c} + \Omega Y_{t-1} + \eta_t, \quad \forall t \in \mathbb{Z} \quad (1.22)$$

où

$$Y_t = \begin{pmatrix} X_t \\ X_{t-1} \\ \vdots \\ X_{t-p+1} \end{pmatrix}, \quad \tilde{c} = \begin{pmatrix} c \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \quad \Omega = \begin{pmatrix} \Phi_1 & \Phi_2 & \dots & \Phi_p \\ I_d & 0 & \dots & 0 \\ 0 & I_d & \dots & 0 \\ & & \ddots & \\ 0 & 0 & \dots & 0 \end{pmatrix}, \quad \eta_t = \begin{pmatrix} \varepsilon_t \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

Remarque 1.3

Y_t, \tilde{c} et η_t sont des vecteurs de dimension $(dp \times 1)$ et la matrice Ω de dimension $(dp \times dp)$ est appelée la matrice compagnon du processus $VAR(p)$. I_d est la matrice d'identité de dimension $(d \times d)$.

Propriété 1.5 Stabilité d'un processus $VAR(p)$

Basée sur l'étude du processus $VAR(1)$, le processus \underline{X}_t est stable si

$$\det(I_{dp} - \Omega z) \neq 0, \quad \text{pour } |z| \leq 1 \quad (1.23)$$

On remarque que

$$\det(I_{dp} - \Omega z) = \det(I_d - \Phi_1 z - \Phi_2 z^2 - \dots - \Phi_p z^p) \quad (1.24)$$

Proposition 1.4 Condition de stabilité d'un d'un processus $VAR(p)$

On montre facilement que X_t peut s'écrire sous la forme :

$$X_t = T Y_t \quad (1.25)$$

où T est une matrice de dimension $(d \times dp)$ qui s'exprime de la façon suivante :

$$T = [I_d : 0 : 0 : \dots : 0] \quad (1.26)$$

Exemple 1.7

Considérons le processus bivarié suivant :

$$X_t = \begin{pmatrix} 0 \\ 1 \end{pmatrix} + \begin{pmatrix} 0.6 & 0.2 \\ 0.3 & 0.7 \end{pmatrix} X_{t-1} + \begin{pmatrix} -0.3 & 0 \\ 0.1 & 0.1 \end{pmatrix} X_{t-2} + \varepsilon_t, \quad \forall t \in \mathbb{Z} \quad (1.27)$$

où $(\varepsilon_t)_{t \in \mathbb{Z}}$ est un bruit blanc indépendant de matrice de covariance

$$\Sigma_\varepsilon = \begin{pmatrix} 0.25 & 0 \\ 0 & 0.09 \end{pmatrix}$$

Le polynôme caractéristique associé est :

$$\begin{aligned}\det(I_2 - \Phi_1 z - \Phi_2 z^2) &= \det \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 0.6 & 0.2 \\ 0.3 & 0.7 \end{bmatrix} z - \begin{bmatrix} -0.3 & 0 \\ 0.1 & 0.1 \end{bmatrix} z^2 \right) \\ &= \det \begin{pmatrix} 1 - 0.6z + 0.3z^2 & -0.2z \\ -0.3z - 0.1z^2 & -0.7z - 0.1z^2 \end{pmatrix}\end{aligned}$$

Pour la forme VAR(1), on a

$$\tilde{c} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \quad \text{et} \quad \Omega = \begin{pmatrix} 0.6 & 0.2 & -0.3 & 0 \\ 0.3 & 0.7 & 0.1 & 0.1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

Donc :

$$Y_t = \tilde{c} + \Omega Y_{t-1} + \eta_t$$

$$\begin{pmatrix} X_{1,t} \\ X_{2,t} \\ X_{1,t-1} \\ X_{2,t-1} \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0.6 & 0.2 & -0.3 & 0 \\ 0.3 & 0.7 & 0.1 & 0.1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} X_{1,t-1} \\ X_{2,t-1} \\ X_{1,t-2} \\ X_{2,t-2} \end{pmatrix} + \begin{pmatrix} \varepsilon_{1,t} \\ \varepsilon_{2,t} \\ 0 \\ 0 \end{pmatrix}$$

```
# Coefficient Omega
Omega = np.array([[0.6,0.2,-0.3,0],[0.3,0.7,0.1,0.1],[1,0,0,0],[0,1,0,0]])
print(Omega)
```

```
## [[ 0.6  0.2 -0.3  0. ]
## [ 0.3  0.7  0.1  0.1]
## [ 1.   0.   0.   0. ]
## [ 0.   1.   0.   0. ]]
```

On calcule les valeurs propres de la matrice Ω .

```
# Valeurs et vecteurs propres de Omega
eigval, eigvect = np.linalg.eig(Omega)
print(eigval.round(4))
```

```
## [ 0.9319+0.j          0.2428+0.4638j  0.2428-0.4638j -0.1175+0.j      ]
```

On calcule les modules associés aux différentes valeurs propres.

```
# Module des valeurs propres
print(abs(eigval))
```

```
## [0.93190023 0.52346508 0.52346508 0.11748337]
```

Les modules respectifs de ces valeurs propres sont tous inférieurs à 1, alors ce processus VAR(2) est stable, donc stationnaire.

Remarque 1.4

Si $(X_t)_{t \in \mathbb{Z}}$ est un processus VAR(p) stable, alors

- Sa moyenne est égale à : $\mathbb{E}(X_t) = T(I_{dp} - \Omega)^{-1} \tilde{c}$
- Sa fonction d'autocovariance est : $\Gamma_X(h) = \text{Cov}(X_t, X_{t-h}) = \sum_{i \geq 0} (T\Omega^{i+h}) \Sigma_\eta (T\Omega^i)'$

Proposition 1.5 Représentation de Wold d'un processus VAR(p)

Le théorème de décomposition de Wold s'applique aussi à un processus VAR(p) stable puisque la stabilité implique la stationnarité. La représentation VAR(1) associée au VAR(p) s'écrit :

$$Y_t = \tilde{c} + \Omega Y_{t-1} + \eta_t \quad (1.28)$$

Sous les conditions de stabilité, le processus $(Y_t)_{t \in \mathbb{Z}}$ a une représentation VMA(∞) suivante :

$$Y_t = m + \sum_{i=0}^{\infty} \Omega^i \eta_{t-i} \quad (1.29)$$

c'est - à - dire que Y_t s'écrit en fonction du passé et du présent des innovations et d'un terme moyen m .

La représentation VMA(∞) de X_t peut être obtenue en prémultipliant l'équation (1.29) par la matrice $T = [I_d : 0 : \dots : 0]$ de format $(d \times dp)$. On a :

$$X_t = Tm + \sum_{i=0}^{\infty} T\Omega^i T' \eta_{t-i} = \mu + \sum_{i=0}^{\infty} \Psi_i \varepsilon_{t-i} \quad (1.30)$$

Ici, $\mu = Tm$, $\Psi_i = T\Omega^i T'$ et dû à la structure spéciale du bruit blanc η_t , nous avons $\eta_t = T' T \eta_t$ et $T \eta_t = \varepsilon_t$. Puisque les Ω^i sont absolument sommables, alors les Ψ_i le sont également.

1.2.2 Fonction d'autocovariance et d'autocorrélation

Proposition 1.6 Fonction d'autocovariance d'un processus VAR(1)

Supposons que $(X_t)_{t \in \mathbb{Z}}$ est un processus VAR(1) stable défini par

$$X_t = c + \Phi_1 X_{t-1} + \varepsilon_t \quad (1.31)$$

où $(\varepsilon_t)_{t \in \mathbb{Z}}$ est bruit blanc multivarié de matrice de covariance Σ_ε .

Sous l'hypothèse de stationnarité de $(X_t)_{t \in \mathbb{Z}}$, les équations de Yule -Walker d'un VAR(1) sont définies comme suit :

$$\Gamma_X(h) = \begin{cases} \Phi_1 (\Gamma_X(1))' + \Sigma_\varepsilon & \text{si } h = 0 \\ \Phi_1 \Gamma_X(h-1) & \text{si } h \geq 1 \end{cases} \quad (1.32)$$

Remarque 1.5

Pour $h = 0$, on a :

$$\Gamma_X(0) = \Phi_1 \Gamma_X(0) \Phi_1' + \Sigma_\varepsilon \quad (1.33)$$

En utilisant l'opérateur vec , on obtient :

$$\text{vec}(\Gamma_X(0)) = (I_{d^2} - \Phi_1 \otimes \Phi_1)^{-1} \text{vec}(\Sigma_\varepsilon) \quad (1.34)$$

En utilisant les équations de Yule - Walker définies précédemment, on a :

$$\Gamma_X(h) = \Phi_1^h \Gamma_X(0), \quad h \geq 0 \quad (1.35)$$

avec

$$\text{vec}(\Gamma_X(0)) = (I_{d^2} - \Phi_1 \otimes \Phi_1)^{-1} \text{vec}(\Sigma_\varepsilon) \quad (1.36)$$

Proposition 1.7 *Autocovariance d'un processus VAR(p) stable*

On considère le processus VAR(p) défini par :

$$X_t = c + \sum_{j=1}^p \Phi_j X_{t-j} + \varepsilon_t \quad (1.37)$$

Les équations de Yule - Walker d'un processus VAR(p) sont définies par

$$\Gamma_X(h) = \begin{cases} \sum_{j=1}^p \Phi_j \Gamma_X'(j) + \Sigma_\varepsilon & \text{si } h = 0 \\ \sum_{j=1}^p \Phi_j \Gamma_X(h-j) & \text{si } h \geq 1 \end{cases} \quad (1.38)$$

Par identification au VAR(1), on a :

$$\Gamma_Y(0) = \Omega \Gamma_Y(0) \Omega' + \Sigma_\eta, \quad \text{où } \Sigma_\eta = \mathbb{E}(\eta_t \eta_t') \quad (1.39)$$

En appliquant l'opérateur vec , on a :

$$\text{vec}(\Gamma_Y(0)) = (I_{(dp)^2} - \Omega \otimes \Omega)^{-1} \text{vec}(\Sigma_\eta) \quad (1.40)$$

Proposition 1.8 *Écriture de $\Gamma_Y(0)$ en fonction de Γ_X*

Si on note par μ_Y la moyenne de $(Y_t)_{t \in \mathbb{Z}}$, alors on a :

$$\begin{aligned}
\Gamma_Y(0) &= \mathbb{E} \left[(Y_t - \mu_Y) (Y_t - \mu_Y)' \right] \\
&= \mathbb{E} \left[\begin{pmatrix} X_t - T(I_{dp} - \Omega)^{-1} \tilde{c} \\ X_{t-1} - T(I_{dp} - \Omega)^{-1} \tilde{c} \\ \vdots \\ X_{t-p+1} - T(I_{dp} - \Omega)^{-1} \tilde{c} \end{pmatrix} \begin{pmatrix} X_t - T(I_{dp} - \Omega)^{-1} \tilde{c} \\ X_{t-1} - T(I_{dp} - \Omega)^{-1} \tilde{c} \\ \vdots \\ X_{t-p+1} - T(I_{dp} - \Omega)^{-1} \tilde{c} \end{pmatrix}' \right] \\
&= \begin{pmatrix} \Gamma_X(0) & \Gamma_X(1) & \dots & \Gamma_X(p-1) \\ \Gamma_X(-1) & \Gamma_X(0) & \dots & \Gamma_X(p-2) \\ & & \ddots & \\ \Gamma_X(-p+1) & \Gamma_X(-p+2) & & \Gamma_X(0) \end{pmatrix} \\
&= \begin{pmatrix} \Gamma_X(0) & \Gamma_X(1) & \dots & \Gamma_X(p-1) \\ \Gamma'_X(1) & \Gamma_X(0) & \dots & \Gamma_X(p-2) \\ & & \ddots & \\ \Gamma'_X(p-1) & \Gamma'_X(p-2) & & \Gamma_X(0) \end{pmatrix}
\end{aligned}$$

Remarque 1.6

On peut utiliser l'équation (1.30) afin de déterminer la moyenne et les autocovariances de X_t , comme suit :

$$\mathbb{E}(X_t) = \mu \quad (1.41)$$

et

$$\begin{aligned}
\Gamma_X(h) &= \mathbb{E} \left[(X_t - \mu)(X_{t-h} - \mu)' \right] = \mathbb{E} \left[\left(\sum_{i=0}^{\infty} \Psi_i \varepsilon_{t-i} \right) \left(\sum_{j=0}^{\infty} \Psi_j \varepsilon_{t-h-j} \right)' \right] \\
&= \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} \Psi_i \underbrace{\mathbb{E}(\varepsilon_{t-i} \varepsilon_{t-h-j}')}_{\substack{\Sigma_\varepsilon \text{ si } i = h+j \\ 0 \text{ si } i \neq h+j}} \Psi_j' = \sum_{i=0}^{\infty} \Psi_i \Sigma_\varepsilon \Psi_{i-h}' \\
&= \sum_{j=0}^{\infty} \Psi_{h+j} \Sigma_\varepsilon \Psi_j'
\end{aligned}$$

On obtient la fonction d'autocovariance à l'ordre h suivante :

$$\Gamma_X(h) = \begin{cases} \sum_{i=0}^{\infty} \Psi_i \Sigma_\varepsilon \Psi_i' & \text{si } h = 0 \\ \sum_{i=0}^{\infty} \Psi_{h+i} \Sigma_\varepsilon \Psi_i' & \text{si } h > 0 \end{cases} \quad (1.42)$$

Les matrices Ψ_i sont déterminées uniquement par $\Psi(z) = \sum_{i=0}^{\infty} \frac{1}{\Phi(z)} \Psi_i z^i$. En effet, on a la relation suivante :

$$\Psi(B)\Phi(B) = I_d \quad (1.43)$$

En prémultipliant le $\Phi(B)X_t = c + \varepsilon_t$ par $\Psi(B)$, on a :

$$X_t = \Psi(B)c + \Psi(B)\varepsilon_t = \left(\sum_{i=0}^{\infty} \Psi_i \right) c + \sum_{i=0}^{\infty} \Psi_i \varepsilon_{t-i} \quad (1.44)$$

Le polynôme $\Psi(B)$ est l'inverse de $\Phi(B)$ et est parfois noté $\Phi(B)^{-1}$ (sous les conditions de stabilité du processus). Les matrices Ψ_i sont obtenues en utilisant la relation suivante :

$$\begin{aligned} I_d &= (\Psi_0 + \Psi_1 B + \Psi_2 B^2 + \dots)(I_d - \Phi_1 B - \Phi_2 B^2 - \dots - \Phi_p B^p) \\ &= \Psi_0 + (\Psi_1 - \Phi_0 \Psi_1) B + (\Psi_2 - \Psi_1 \Phi_1 - \Psi_0 \Phi_2) B^2 + \dots \\ &\quad + \left(\Psi_i - \sum_{j=1}^{j=i} \Psi_{i-j} \Phi_j \right) B^i + \dots \end{aligned}$$

Or

$$\begin{aligned} I_d &= \Psi_0 \\ 0 &= \Psi_1 - \Psi_0 \Phi_1 \\ 0 &= \Psi_2 - \Psi_1 \Phi_1 - \Psi_0 \Phi_2 \\ &\vdots \\ 0 &= \Psi_i - \sum_{j=1}^{j=i} \Psi_{i-j} \Phi_j \\ &\vdots \end{aligned}$$

avec $\Phi_j = 0$ pour $j > p$. Ainsi, on a la relation récursive suivante :

$$\Psi_i = \begin{cases} I_d & \text{si } i = 0 \\ \sum_{j=1}^{j=i} \Psi_{i-j} \Phi_j & \text{si } i > 0 \end{cases} \quad (1.45)$$

Proposition 1.9 *Autocorrélation d'un processus VAR(p) stable*

Soit D une matrice diagonale dont les éléments diagonaux sont les racines carrées des éléments diagonaux de $\Gamma_X(0)$. L'autocorrélation d'un processus VAR(p) stable, noté $R_X(\cdot)$ est définie par :

$$R_X(h) = D^{-1} \Gamma_X(h) D^{-1} \quad (1.46)$$

avec

$$D^{-1} = \begin{pmatrix} 1/\sqrt{\gamma_{1,1}(0)} & \dots & 0 \\ \vdots & \dots & \vdots \\ 0 & \dots & 1/\sqrt{\gamma_{d,d}(0)} \end{pmatrix} \quad (1.47)$$

où $\gamma_{i,j}(h) = \text{Cov}(X_{i,t}, X_{j,t-h})$

Remarque 1.7

L'autocorrélation entre $X_{i,t}$ et $X_{j,t-h}$ est

$$\rho_{ij}(h) = \frac{\gamma_{ij}(h)}{\sqrt{\gamma_{ii}(0)}\sqrt{\gamma_{jj}(0)}} \quad (1.48)$$

Exemple 1.8

Considérons le processus VAR(1) bi-dimensionnel défini par :

$$X_t = \begin{pmatrix} 5 \\ 3 \end{pmatrix} + \begin{pmatrix} 0.2 & 0.3 \\ -0.6 & 1.1 \end{pmatrix} X_{t-1} + \varepsilon_t \quad (1.49)$$

avec $(\varepsilon_t)_{t \in \mathbb{Z}}$ un bruit blanc bidimensionnel avec matrice de covariance

$$\Sigma_\varepsilon = \begin{pmatrix} 1 & 0.8 \\ 0.8 & 2 \end{pmatrix} \quad (1.50)$$

— Calculer les valeurs propres de la matrice autorégressive Φ_1 . Est-il stable ?

```
# VAR(1) - bidimensionnel
Phi1 = np.array([[0.2, 0.3], [-0.6, 1.1]])
c = np.array([[5], [3]])
sigma = np.array([[1, 0.8], [0.8, 2]])
# Valeurs et vecteurs propres de Phi1
eigval, eigvect = np.linalg.eig(Phi1)
print(eigval)
```

```
## [0.5 0.8]
```

Les valeurs propres de Φ_1 sont 0.5 et 0.8. Les modules respectifs de ces valeurs propres sont 0.5 et 0.8. Comme ils sont tous inférieurs à 1, alors ce processus VAR(1) est stable, donc stationnaire.

— Calculer $\Gamma_X(0)$, $\Gamma_X(1)$, $\Gamma_X(2)$ et $\Gamma_X(3)$

On commence par calculer $\text{vec}(\Gamma_X(0))$

```
# Calcul des autocovariances
vecsigma = vec(sigma)
PhikronPhi1 = np.kron(Phi1, Phi1)
inv = inv_ident_minus_matrix(PhikronPhi1) # calcul de (I-Phi1)^(-1)
vecgamma0 = inv.dot(vecsigma)
print(vecgamma0)
```

```
## [[2.28888889]
## [3.51111111]
## [3.51111111]
## [8.62222222]]
```

On redéfinit les dimensions du vecteur afin d'obtenir la matrice $\Gamma_X(0)$.

```
# Calcul de gamma 0
gamma0 = vecgamma0.reshape((Phi1.shape[0], Phi1.shape[1]))
print(gamma0)
```

```
## [[2.28888889 3.51111111]
##   [3.51111111 8.62222222]]
```

On applique la formule suivante pour les autres autocovariances :

$$\Gamma_X(h) = \Phi_1^h \Gamma_X(0) = \Phi_1 \Gamma_X(h-1) \quad \text{pour } h \geq 1 \quad (1.51)$$

Pour $h = 1$, on obtient :

```
# Calcul de gamma1
gamma1 = Phi1.dot(gamma0)
print(gamma1)

## [[1.51111111 3.28888889]
##   [2.48888889 7.37777778]]
```

Pour $h = 2$, on obtient :

```
# Calcul de gamma2
gamma2 = Phi1.dot(gamma1)
print(gamma2)

## [[1.04888889 2.87111111]
##   [1.83111111 6.14222222]]
```

Pour $h = 3$, on obtient :

```
# Calcul de gamma3
gamma3 = Phi1.dot(gamma2)
print(gamma3)

## [[0.75911111 2.41688889]
##   [1.38488889 5.03377778]]
```

— En déduire $R_X(0), R_X(1), R_X(2)$ et $R_X(3)$

On applique la formule :

$$R_X(h) = D^{-1} \Gamma_X(h) D^{-1} \quad (1.52)$$

Commençons par calculer D^{-1} .

```
# Matrice D et D^(-1)
D = np.diag(np.sqrt(np.diag(gamma0)))
invD = np.linalg.inv(D)
print(invD)
```

```
## [[0.66097897 0.          ]
##  [0.          0.34055746]]
```

Pour $h = 0$, on obtient :

```
# Calcul de R(0)
R0 = invD.dot(gamma0.dot(invD))
print(R0)
```

```
## [[1.          0.79035574]
##  [0.79035574 1.          ]]
```

Pour $h = 1$, on obtient :

```
# Calcul de R(1)
R1 = invD.dot(gamma1.dot(invD))
print(R1)
```

```
## [[0.66019417 0.74033322]
##  [0.56025217 0.8556701 ]]
```

Pour $h = 2$, on obtient :

```
# Calcul de R(2)
R2 = invD.dot(gamma2.dot(invD))
print(R2)
```

```
## [[0.45825243 0.6462909 ]
##  [0.41218553 0.71237113]]
```

Pour $h = 3$, on obtient :

```
# Calcul de R(3)
R3 = invD.dot(gamma3.dot(invD))
print(R3)
```

```
## [[0.33165049 0.54404488]
##  [0.31174031 0.58381443]]
```

Exemple 1.9

Considérons le processus bivarié suivant :

$$X_t = \begin{pmatrix} -3 \\ 1 \end{pmatrix} + \begin{pmatrix} 1/6 & 0 \\ -1/5 & \sqrt{2}/2 \end{pmatrix} X_{t-1} + \begin{pmatrix} 0 & 0 \\ 1/3 & -1/8 \end{pmatrix} X_{t-2} + \varepsilon_t \quad (1.53)$$

où $(\varepsilon_t)_{t \in \mathbb{Z}}$ est un bruit blanc bivarié de matrice de covariance :

$$\Sigma_\varepsilon = \begin{pmatrix} 1/4 & 0 \\ 0 & 1/8 \end{pmatrix} \quad (1.54)$$

— Étudier la stabilité du processus $(X_t)_{t \in \mathbb{Z}}$

L'écriture VAR(1) du VAR(2) précédent est :

$$Y_t = \begin{pmatrix} X_t \\ X_{t-1} \end{pmatrix} = \begin{pmatrix} -3 \\ 1 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 1/6 & 0 & 0 & 0 \\ -1/5 & \sqrt{2}/2 & 1/3 & -1/8 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} X_{t-1} \\ X_{t-2} \end{pmatrix} + \begin{pmatrix} \varepsilon_t \\ 0 \end{pmatrix} \quad (1.55)$$

```
# Matrice Omega
Omega = np.array([[1/6,0,0,0],[-1/5,np.sqrt(2)/2, 1/3, -1/8],
                  [1, 0, 0, 0], [0, 1, 0 ,0]])
# Valeurs et vecteurs propres de Omega
eigval, eigvect = np.linalg.eig(Omega)
print(eigval)
```

```
## [0.35355339 0.35355339 0.          0.16666667]
```

Les modules respectifs de ces valeurs propres sont tous inférieurs à 1, alors ce processus VAR(2) est stable, donc stationnaire.

— Calcul de $\Gamma_X(0), \Gamma_X(1), \Gamma_X(2)$ et $\Gamma_X(3)$

On calcule

$$\text{vec}[\Gamma_Y(0)] = (I_{16} - \Omega \otimes \Omega)^{-1} \text{vec}[\Sigma_\eta] \quad (1.56)$$

```
# Matrice sigma
sigma = np.array([[1/4,0,0,0],[0,1/8,0,0],[0,0,0,0],[0,0,0,0]])
print(sigma)
```

```
## [[0.25  0.    0.    0.   ]
##   [0.    0.125 0.    0.   ]
##   [0.    0.    0.    0.   ]
##   [0.    0.    0.    0.   ]]
```

```
# Autocovariance
vecsigma = vec(sigma)
OmegakronOmega = np.kron(Omega,Omega)
inv = inv_ident_minus_matrix(OmegakronOmega)
vecgamma0 = inv.dot(vecsigma)
print(vecgamma0)
```

```
## [[ 0.25714286]
##   [-0.00698998]
##   [ 0.04285714]
##   [-0.001165 ]
##   [-0.00698998]
##   [ 0.23988902]
##   [-0.0419399 ]
##   [ 0.13959572]
```

```
## [ 0.04285714]
## [-0.0419399 ]
## [ 0.25714286]
## [-0.00698998]
## [-0.001165  ]
## [ 0.13959572]
## [-0.00698998]
## [ 0.23988902]]
```

On transforme $\text{vec}[\Gamma_Y(0)]$ sous forme de matrice de format 4×4

$$\Gamma_Y(0) = \begin{pmatrix} \Gamma_X(0) & \Gamma_X(1) \\ \Gamma_X(1)' & \Gamma_X(0) \end{pmatrix} \quad (1.57)$$

```
# Transformation en matrix de format (4,4)
G = vecgamma0.reshape((Omega.shape[0], Omega.shape[1]))
print(G)

## [[ 0.25714286 -0.00698998  0.04285714 -0.001165  ]
## [-0.00698998  0.23988902 -0.0419399  0.13959572]
## [ 0.04285714 -0.0419399  0.25714286 -0.00698998]
## [-0.001165    0.13959572 -0.00698998  0.23988902]]
```

On extrait les matrices $\Gamma_X(0)$

```
# Extraction de gamma0
gamma0 = G[:, :2]
print(gamma0)

## [[ 0.25714286 -0.00698998]
## [-0.00698998  0.23988902]]
```

... et $\Gamma_X(1)$

```
# Extraction de gamma1
gamma1 = G[:, 2:]
print(gamma1)

## [[ 0.04285714 -0.001165  ]
## [-0.0419399  0.13959572]]
```

On calcule ensuite

$$\Gamma_X(2) = \Phi_1 \Gamma_X(1) + \Phi_2 \Gamma_X(0) \quad (1.58)$$

```
# Matrice Phi1 et Phi2
Phi1 = np.array([[1/6, 0], [-1/5, np.sqrt(2)/2]])
Phi2 = np.array([[0, 0], [1/3, -1/8]])
# Calcul de gamma2
gamma2 = Phi1.dot(gamma1) + Phi2.dot(gamma0)
print(gamma2)
```

```
## [[ 0.00714286 -0.00019417]
##  [ 0.04836062  0.06662596]]
```

... et

$$\Gamma_X(3) = \Phi_1 \Gamma_X(2) + \Phi_2 \Gamma_X(1) \quad (1.59)$$

```
# Calcul de gamma3
gamma3 = Phi1.dot(gamma2) + Phi2.dot(gamma1)
print(gamma3)
```

```
## [[ 1.19047619e-03 -3.23610314e-05]
##  [ 5.22957520e-02  2.93127023e-02]]
```

1.2.3 Fonction d'autocorrélation partielle et densité spectrale

1.2.3.1 Autocorrélation partielle d'un VAR(p)

Dans le cas univarié, pour identifier le nombre de retards p d'un processus AR on utilise la fonction d'autocorrélation partielle. En multivariée, on dispose de matrice d'autocorrélations partielles. Cependant, celles-ci sont très difficiles à calculer, en général on ne les utilise pas. En pratique, on impose la forme *a priori* en procédant par tâtonnement. L'idée est de retenir des ordres de retards p suffisamment grands puis d'en déduire la taille à l'aide de tests. On choisit ainsi en général un VAR(4) pour des données trimestrielles, un VAR(12) ou VAR(8) pour des données mensuelles, etc. On verra cependant qu'il existe des outils, notamment les critères d'information, qui évitent de fixer arbitrairement la valeur de p afin que le nombre de paramètres à estimer pd^2 en soit pas trop grand. De même, les tests de causalité vont permettre de hiérarchiser les variables et de réduire le nombre de paramètres à estimer.

1.2.3.2 Densité spectrale d'un VAR(P)

Dans le cas d'un processus VAR, la densité spectrale est donnée par :

$$F(\omega) = \frac{1}{2\pi} \Phi^{-1}(e^{-i\omega}) \Sigma_\varepsilon \Phi^{-1}(e^{i\omega})' \quad (1.60)$$

On rappelle que cette fonction fournit la même information que la fonction d'autocovariance.

Estimation et validation d'un processus VAR

Sommaire

2.1 Estimation des paramètres d'un VAR(p)	27
2.2 Détermination du nombre de retards	40
2.3 Validation de modèle et prévision	43

Dans ce chapitre, on considère un processus stochastique X_t à d ($d \geq 2$) dimensions dont les composantes sont $X_{1,t}, \dots, X_{d,t}$ tel que $X_t = (X_{1,t}, \dots, X_{d,t})'$. On suppose également que ce processus est généré par un processus vectoriel autoregressif d'ordre p , c'est-à-dire que :

$$X_t = c + \Phi_1 X_{t-1} + \dots + \Phi_p X_{t-p} + \varepsilon_t \quad (2.1)$$

où ε_t est un bruit blanc multivarié de matrice de covariance Σ_ε non singulière. Les matrices $\Phi_j, j = 1, \dots, p$ de format d^2 et le vecteur c de format $(d \times 1)$ sont inconnus et coefficients seront estimés à partir d'une suite de réalisation X_1, \dots, X_T du processus X_t .

2.1 Estimation des paramètres d'un VAR(p)

Supposons X_1, \dots, X_T un échantillon de taille T , d'un processus d -vectoriel X_t . Nous allons aussi supposer que nous avons p valeurs initiales X_{-p+1}, \dots, X_0 . Nous supposons donc que l'ordre p de notre processus est connu d'avance. Mais en pratique, les paramètres d'un VAR(p) ne sont pas connus. Ces paramètres sont les coefficients de régression, c'est-à-dire, le vecteur de constantes c et les éléments des matrices Φ_1, \dots, Φ_p . Il faut donc les estimer. On peut être aussi intéressé à estimer la matrice des variances covariances des bruits blancs vectoriels appelée Σ_ε . Différentes méthodes existent. Celles que nous allons présenter sont la méthode des moindres carrés généralisés, la méthode du maximum de vraisemblance et la méthode Yule - Walker.

2.1.1 Estimation par moindres carrés généralisés

La méthode des moindres carrés généralisés (GLS, *Generalized Least Squares*) permet d'estimer les paramètres du modèle étudié en minimisant la somme pondérée des carrés résiduels. Pour définir cette méthode, nous allons utiliser certaines notations. Soit $X = (X_1, \dots, X_T)$ de dimensions $(d \times T)$. Sur la j -ième ligne de cette matrice, se trouvent les n observations de la j -ième composante du processus d -vectoriel. Une ligne de sa transposée X' est alors une observation

de notre processus. En d'autres termes, on trouve la première observation des d composantes sur la première ligne de cette matrice X' , et ainsi de suite. Soit $B = (c, \Phi_1, \dots, \Phi_p)$ de dimensions $(d \times (dp + 1))$, la matrice des coefficients de régression

$$Z_t = \begin{pmatrix} 1 \\ X_t \\ \vdots \\ X_{t-p+1} \end{pmatrix} \quad (2.2)$$

de dimensions $((dp + 1) \times 1)$, $Z = (Z_0, \dots, Z_{T-1})$ de dimensions $((dp + 1) \times T)$. C'est l'ensemble des p premières valeurs de l'échantillon tiré. Soit $E = (\varepsilon_1, \dots, \varepsilon_T)$ de dimension $(d \times T)$, la matrice des bruits blancs vectoriels associés à chaque observation de notre échantillon. On peut donc écrire le processus sous la forme :

$$X = BZ + E \quad (2.3)$$

En appliquant la fonction vec , l'équation précédente devient :

$$\text{vec}(X) = \text{vec}(BZ) + \text{vec}(E) = \text{vec}(I_d BZ) + \text{vec}(E)$$

On en déduit :

$$\text{vec}(X) = (Z' \otimes I_d) \text{vec}(B) + \text{vec}(E) \quad (2.4)$$

Si on note $x = \text{vec}(X)$, $\beta = \text{vec}(B)$ et $e = \text{vec}(E)$, des matrices de dimension $(dT + 1)$, $((d^2p + d) \times 1)$ et $((d^2p + d) \times 1)$ respectivement, on peut écrire une formule équivalente à (2.4) qui est :

$$x = (Z' \otimes I_d) \beta + e \quad (2.5)$$

On montre aisément que la matrice de covariance de e est

$$\Sigma_e = I_T \otimes \Sigma_\varepsilon \quad (2.6)$$

2.1.1.1 Estimateur des moindres carrés généralisés

L'estimateur du vecteur β par la méthode des moindres carrés généralisée est la valeur qui minimise $l(\beta)$ donnée par :

$$\begin{aligned} l(\beta) &= e' \Sigma_e^{-1} e = e' (I_T \otimes \Sigma_\varepsilon)^{-1} e = e' (I_T \otimes \Sigma_\varepsilon^{-1}) e \\ &= [x - (Z' \otimes I_d) \beta]' (I_T \otimes \Sigma_\varepsilon^{-1}) [x - (Z' \otimes I_d) \beta] \\ &= \text{vec}(X - BZ)' (I_T \otimes \Sigma_\varepsilon^{-1}) \text{vec}(X - BZ) \\ &= \text{tr} [(X - BZ)' \Sigma_\varepsilon^{-1} (X - BZ)] \end{aligned}$$

où $\text{tr}(A)$ représente la trace d'une matrice A . On peut montrer aisément que

$$\begin{aligned}
l(\beta) &= x' (I_T \otimes \Sigma_\varepsilon^{-1}) x + \beta' (Z \otimes I_d) (I_T \otimes \Sigma_\varepsilon^{-1}) (Z' \otimes I_d) \beta - 2\beta' (Z \otimes I_d) (I_T \otimes \Sigma_\varepsilon^{-1}) x \\
&= x' (I_T \otimes \Sigma_\varepsilon^{-1}) x + \beta' (ZZ' \otimes \Sigma_\varepsilon^{-1}) \beta - 2\beta' (Z \otimes \Sigma_\varepsilon^{-1}) x
\end{aligned}$$

Après avoir dérivé $l(\beta)$ par rapport à β , on a :

$$\frac{\partial}{\partial \beta} l(\beta) = 2 (ZZ' \otimes \Sigma_\varepsilon^{-1}) \beta - 2 (Z \otimes \Sigma_\varepsilon^{-1}) x$$

En égalant cette dérivée à 0, l'estimateur des moindres carrés généralisés (GLS) de β satisfait les équations normales :

$$(ZZ' \otimes \Sigma_\varepsilon^{-1}) \hat{\beta} = (Z \otimes \Sigma_\varepsilon^{-1}) x \quad (2.7)$$

Ce qui implique que :

$$\hat{\beta} = (ZZ' \otimes \Sigma_\varepsilon^{-1})^{-1} (Z \otimes \Sigma_\varepsilon^{-1}) x = [(ZZ')^{-1} \otimes \Sigma_\varepsilon] (Z \otimes \Sigma_\varepsilon^{-1}) x = [(ZZ')^{-1} Z \otimes I_d] x$$

L'estimateur des moindres carrés généralisés (GLS) de β est

$$\hat{\beta} = \text{vec}(B) = [(ZZ')^{-1} Z \otimes I_d] \text{vec}(X) \quad (2.8)$$

Comme la matrice hessienne de $l(\beta)$ qui est égale à

$$\frac{\partial^2}{\partial \beta \partial \beta'} l(\beta) = 2(ZZ' \otimes \Sigma_\varepsilon^{-1}) \quad (2.9)$$

est définie positive, alors l'estimateur obtenu $\hat{\beta}$ est bel et bien un minimum. (ces résultats sont valides si la matrice ZZ' est non singulière).

Dans le cas du modèle VAR(p), L'estimateur des moindres carrés généralisé est identique à l'estimateur des moindres carrés ordinaires obtenu par minimisation de la fonction $\tilde{l}(\beta)$ définie comme suit :

$$\tilde{l}(\beta) = e' e = [x - (Z' \otimes I_d) \beta]' [x - (Z' \otimes I_d) \beta] \quad (2.10)$$

Ce résultat est du à Zellner (1962) qui a montré que la méthode des moindres carrés généralisés et la méthode des moindres carrés ordinaires dans un modèle à équation multiple sont identiques si les régresseurs dans toutes les équations sont les mêmes.

L'estimateur des moindres carrés généralisé peut être écrit d'une manière différente :

$$\hat{\beta} = [(ZZ')^{-1} Z \otimes I_d] [(Z' \otimes I_d) \beta + e] = \beta + ((ZZ')^{-1} Z \otimes I_d) e \quad (2.11)$$

On peut écrire de façon équivalente à (2.8), sans la fonction vec , que l'estimateur des moindres carrés généralisé de B , noté \hat{B} , est

$$\text{vec}(\widehat{B}) = \widehat{\beta} = \left[(ZZ')^{-1} Z \otimes I_d \right] \text{vec}(X) = \text{vec}(XZ'(ZZ')^{-1})$$

Ainsi, on a l'écriture suivante :

$$\widehat{B} = XZ'(ZZ')^{-1} = (BZ + E)Z'(ZZ')^{-1} = B + EZ'(ZZ')^{-1} \quad (2.12)$$

Un estimateur des moindres carrés généralisé de $\Sigma_\varepsilon = \mathbb{E}[\varepsilon_t \varepsilon_t']$ découlant de celui de B est :

$$\widetilde{\Sigma}_\varepsilon = \frac{1}{T} \sum_{t=1}^{t=T} \widehat{\varepsilon}_t \widehat{\varepsilon}_t' = \frac{1}{T} X \left(I_T - Z'(ZZ')^{-1} Z \right) X' \quad (2.13)$$

preuve

Puisque $\Sigma_\varepsilon = \mathbb{E}(\varepsilon_t \varepsilon_t')$, alors un estimateur plausible de la matrice de variance covariance est donné par :

$$\begin{aligned} \widetilde{\Sigma}_\varepsilon &= \frac{1}{T} \sum_{t=1}^{t=T} \widehat{\varepsilon}_t \widehat{\varepsilon}_t' = \frac{1}{T} \widehat{E} \widehat{E}' = \frac{1}{T} (X - \widehat{B}Z)(X - \widehat{B}Z)' \\ &= \frac{1}{T} \left[X - XZ'(ZZ')^{-1}Z \right] \left[X - XZ'(ZZ')^{-1}Z \right]' \\ &= \frac{1}{T} X \left(I_T - Z'(ZZ')^{-1}Z \right) X' \end{aligned}$$

On peut aussi définir un autre estimateur non biaisé de Σ_ε qui est :

$$\widehat{\Sigma}_\varepsilon = \frac{1}{T - dp - 1} X \left(I_T - Z'(ZZ')^{-1}Z \right) X' = \frac{T}{T - dp - 1} \widetilde{\Sigma}_\varepsilon \quad (2.14)$$

2.1.1.2 Propriétés de l'estimateur GLS

Nous étudions à présent les propriétés asymptotiques de l'estimateur des moindres carrés généralisés de notre processus d -vectoriel sous le postulat qu'il est stable. De plus, son bruit blanc vectoriel doit être indépendant et ses 4 premiers moments doivent exister et être bornés. Il faut noter que les bruits blancs vectoriels gaussiens remplissent ces conditions.

Propriété 2.1 Propriétés asymptotiques de \widehat{B}

Si X_t est un processus stable et $\widehat{B} = XZ'(ZZ')^{-1}$ est l'estimateur des moindres carrés généralisés de B

$$\frac{ZZ'}{T} \xrightarrow{\mathbb{P}} \Gamma, \quad T \rightarrow +\infty \quad (2.15)$$

existe et est non singulière. De plus,

$$\frac{1}{\sqrt{T}} \sum_{t=1}^T \text{vec}(\varepsilon_t Z_{t-1}') = \frac{1}{\sqrt{T}} \text{vec}(EZ') = \frac{1}{\sqrt{T}} (Z \otimes I_d) e \xrightarrow{\mathcal{L}} \mathcal{N}(0, \Gamma \otimes \Sigma_\varepsilon), \quad T \rightarrow +\infty \quad (2.16)$$

et sous les conditions énoncées sur les bruits blancs vectoriels de X_t , on a :

$$\widehat{B} \xrightarrow{\mathbb{P}} B, \quad T \rightarrow +\infty \quad (2.17)$$

De plus,

$$\sqrt{T}(\widehat{\beta} - \beta) \xrightarrow{\mathcal{L}} \mathcal{N}(0, \Gamma^{-1} \otimes \Sigma_\varepsilon) \quad (2.18)$$

Ces propriétés permettent de faire l'inférence sur le paramètre β comme calculer les intervalles de confiance pour β ou procéder à des tests d'hypothèses comme les tests de Student et de Fisher.

Propriété 2.2 *Propriétés asymptotiques de $\widehat{\Sigma}_\varepsilon$ et $\widetilde{\Sigma}_\varepsilon$*

Si X_t est un processus stable et $\widehat{B} = XZ'(ZZ')^{-1}$ est l'estimateur des moindres carrés généralisée de B , et si

$$\frac{ZZ'}{T} \xrightarrow{\mathbb{P}} \Gamma, \quad \text{quand } T \rightarrow +\infty \quad (2.19)$$

existe et est non singulière. De plus,

$$\frac{1}{T} \sum_{t=1}^{t=T} \text{vec}(\varepsilon_t Z'_{t-1}) \xrightarrow{\mathcal{L}} \mathcal{N}(0, \Gamma \otimes \Sigma_\varepsilon), \quad \text{quand } T \rightarrow +\infty \quad (2.20)$$

et sous les conditions énoncées sur les bruits blancs vectoriels de X_t , on a :

$$\widehat{\Sigma}_\varepsilon \xrightarrow{\mathbb{P}} \Sigma_\varepsilon \quad \text{et} \quad \widetilde{\Sigma}_\varepsilon \xrightarrow{\mathbb{P}} \Sigma_\varepsilon, \quad \text{quand } T \rightarrow +\infty \quad (2.21)$$

Exemple 2.1 *Application données réelles*

Nous illustrons l'estimation des paramètres du processus vectoriel autorégressif avec le jeu de données extrait du livre de Lütkepohl (2005). Ce jeu de données est composé de 3 variables macroéconomiques : le taux d'investissement (*invest*), le revenu national (*income*) et la consommation (*conso*). Les données ont été observées du premier trimestre 1960 au quatrième trimestre 1982, soit 92 observations.

```
# Chargement des données
import statsmodels.api as sm
import matplotlib.pyplot as plt
dta = sm.datasets.webuse('lutkepohl2', 'https://www.stata-press.com/data/r12/')
dta.index = dta.qtr
dta.index.freq = dta.index.inferred_freq
df = dta.loc['1960-01-01':, ["inv", "inc", "consump"]]
# Représentation graphique
fig, axe = plt.subplots(figsize=(16,6))
df.plot(ax=axe);
axe.set_xlabel('Date');
axe.set_ylabel('$X_{\{t\}}$');
plt.show()
```

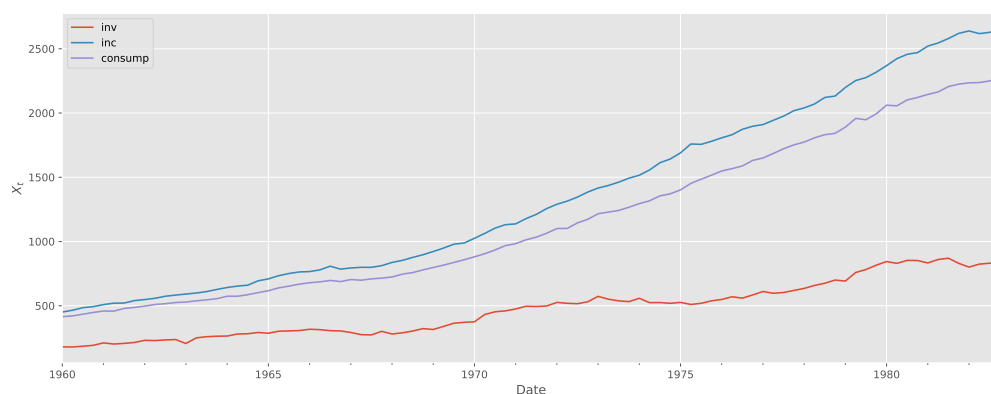


Figure 2.1 – Visualisation des séries investissement/Revenu/consommation

La figure (2.1) montre que les séries possèdent une tendance, par conséquent elles sont considérées non stationnaires. Nous effectuons une double transformation sur les séries afin de les rendre stationnaires : premièrement en effectuant une transformation logarithmique et deuxièmement en appliquant un filtre différencié.

```
endog = dta.loc["1960-04-01":, ["dln_inv", "dln_inc", "dln_consump"]]
# Représentation graphique
fig, axe = plt.subplots(3,1,figsize=(16,8))
for idx,name in enumerate(endog.columns):
    endog[name].plot(ax=axe[idx],color = "black");
    axe[idx].set_xlabel("année");
    axe[idx].set_ylabel("valeur");
    axe[idx].set_title(name);
plt.tight_layout();
plt.show()
```

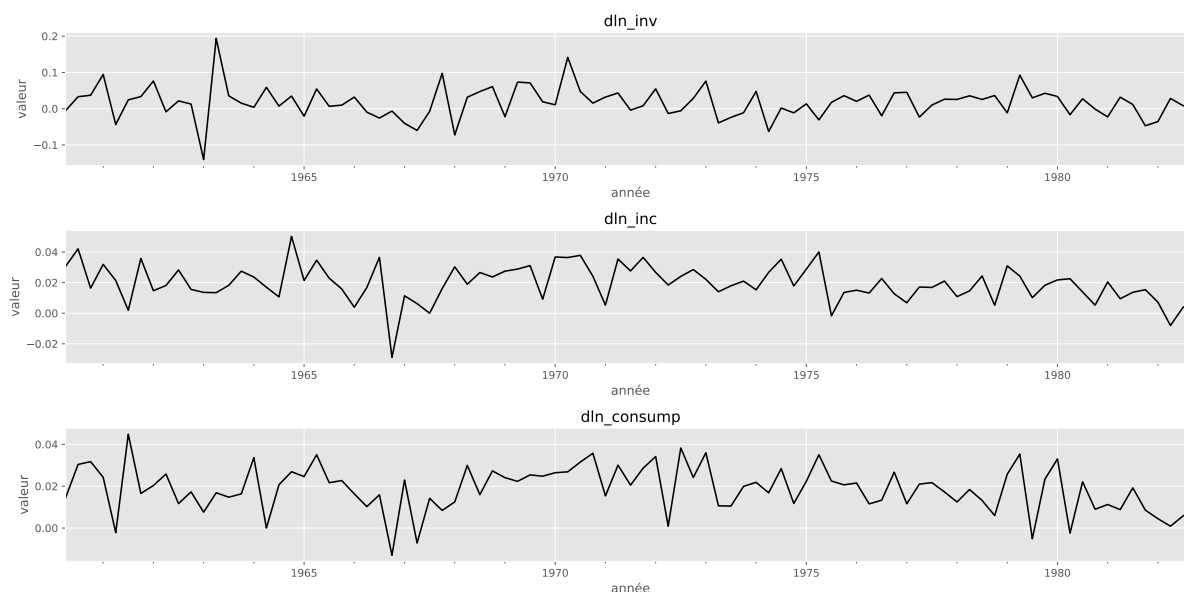


Figure 2.2 – First differences of logarithms of West German investment, income and consumption

Nous utilisons les données de 1960 à 1978 en apprentissage et les données de 1979 à 1982 en test.

On suppose que nos données sont générées par processus $VAR(2)$. Le choix de 2 est arbitraire à ce niveau.

Sous Python, l'estimation des paramètres d'un modèle VAR(p) s'effectue avec la fonction `VAR` de la librairie Statsmodels.

```
# Subdivision
DTrain = dta.loc["1960-04-01":"1978-10-01",["dln_inv","dln_inc","dln_consump"]]
# Estimation des paramètres
from statsmodels.tsa.api import VAR
model = VAR(DTrain)
results = model.fit(2)
var_params = results.params.round(3).T
```

Table 2.1 – Coefficients du modèle VAR(2)

	const	L1.dln_inv	L1.dln_inc	L1.dln_consump	L2.dln_inv	L2.dln_inc	L2.dln_consump
dln_inv	-0.017	-0.320	0.146	0.961	-0.161	0.115	0.934
dln_inc	0.016	0.044	-0.153	0.288	0.050	0.019	-0.010
dln_consump	0.013	-0.002	0.225	-0.264	0.034	0.355	-0.022

Notre modèle s'écrit donc :

$$\begin{pmatrix} \Delta \log(inv)_t \\ \Delta \log(inc)_t \\ \Delta \log(consump)_t \end{pmatrix} = \begin{pmatrix} -0.017 \\ 0.016 \\ 0.013 \end{pmatrix} + \begin{pmatrix} -0.320 & 0.146 & 0.961 \\ 0.044 & -0.153 & 0.289 \\ -0.002 & 0.225 & -0.264 \end{pmatrix} \begin{pmatrix} \Delta \log(inv)_{t-1} \\ \Delta \log(inc)_{t-1} \\ \Delta \log(consump)_{t-1} \end{pmatrix} \\ + \begin{pmatrix} -0.161 & 0.115 & 0.934 \\ 0.050 & 0.019 & -0.010 \\ 0.034 & 0.355 & -0.022 \end{pmatrix} \begin{pmatrix} \Delta \log(inv)_{t-2} \\ \Delta \log(inc)_{t-2} \\ \Delta \log(consump)_{t-2} \end{pmatrix} + \begin{pmatrix} \hat{\varepsilon}_{1,t} \\ \hat{\varepsilon}_{2,t} \\ \hat{\varepsilon}_{3,t} \end{pmatrix} \quad (2.22)$$

où la matrice de variance - covariance des résidus peut être estimée par :

$$\hat{\Sigma}_\varepsilon = \frac{1}{T - dp - 1} \hat{\varepsilon} \hat{\varepsilon}' \quad (2.23)$$

```
# Matrice de variance - covariance des résidus
Omega_sigma = results.resid.T.dot(results.resid)/results.df_resid
print(Omega_sigma)
```

```
##          dln_inv  dln_inc  dln_consump
## dln_inv    0.002130  0.000072    0.000123
## dln_inc    0.000072  0.000137    0.000061
## dln_consump 0.000123  0.000061    0.000089
```

L'attribut `res.sigma_u` nous renvoie la matrice des variances - covariances des résidus.

```
# Matrice de variance - covariance des résidus
print(results.sigma_u)
```

```
##          dln_inv    dln_inc    dln_consump
## dln_inv    0.002130  0.000072    0.000123
## dln_inc    0.000072  0.000137    0.000061
## dln_consump 0.000123  0.000061    0.000089
```

Les résultats sont identiques.

On vérifie si notre processus estimé est stable, c'est-à-dire qu'on détermine les racines du polynôme $\det(I_3 - \hat{\Phi}_1 z - \hat{\Phi}_2 z^2)$. L'attribut `res.roots` de notre modèle nous renvoie les racines du polynôme.

```
# Racines du polynôme
print(results.roots.round(2))

## [-2.69-0.j   -0.32+2.01j -0.32-2.01j -1.29+1.28j -1.29-1.28j  1.75-0.j   ]
```

Nous avons deux racines réelles et quatre racines complexes.

```
# Module des racines
print(abs(results.roots))

## [2.69398226  2.03367606  2.03367606  1.81398677  1.81398677  1.75293821]
```

Toutes les racines sont de module supérieur à 1, donc la condition de stabilité du VAR(2) est vérifiée.

```
# Application au modèle ARMA(1,2)
fig, axe = plt.subplots(figsize=(10,10))
plotroot(results.roots,title = "VAR(2) roots", ax = axe);
plt.show()
```

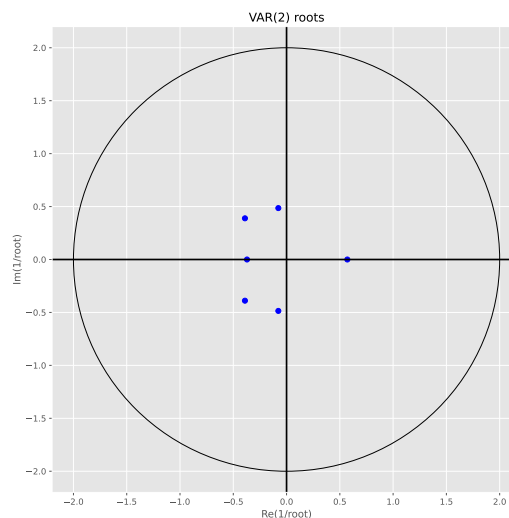


Figure 2.3 – Racines caractéristiques inverses du modèle VAR(2) ajustée

Une autre approche pour vérifier la stabilité de notre modèle est d'utiliser la méthode `res.is_stable()` basée sur les coefficients du modèle. Elle retourne un booléen : `True` si le modèle est stable ou `False` si le modèle est instable.

```
# Stabilité du modèle VAR(2)
print(results.is_stable())
```

```
## True
```

Nous ne nous attardons pas sur la significativité des paramètres, cela se fera ultérieurement.

La matrice d'autocovariance à l'ordre h , $\hat{\Gamma}_X(h)$ est obtenue grâce à la méthode `res.sample_acov()`.

```
# Autocovariance à l'ordre 0
print(results.sample_acov(nlags=0))

## [[2.20947154e-03 5.11163635e-05 1.27429914e-04]
##    [5.11163635e-05 1.40175020e-04 5.56711202e-05]
##    [1.27429914e-04 5.56711202e-05 1.07717664e-04]]]
```

et la matrice d'autocorrélation $\hat{R}_X(h)$ est obtenue grâce à la méthode `res.sample_acorr()`.

```
# Autocorrélation à l'ordre 0
print(results.sample_acorr(nlags=0))

## [[1.          0.09185027 0.26120633]
##    [0.09185027 1.0000001  0.45305526]
##    [0.26120633 0.45305526 1.0000001  ]]
```

2.1.2 Estimation des moindres carrés avec données ajustées à la moyenne

2.1.2.1 Estimation lorsque la moyenne du processus est connue

Occasionnellement, un processus VAR(p) s'écrit sous la forme ajustée à la moyenne suivante :

$$(X_t - \mu) = \Phi_1(X_{t-1} - \mu) + \dots + \Phi_p(X_{t-p} - \mu) + \varepsilon_t \quad (2.24)$$

L'estimateur des moindres carrés de ce modèle est simple si la moyenne μ du processus est connue. On définit :

$$\begin{cases} X^0 := (X_1 - \mu, \dots, X_{T-1} - \mu) & (d \times T) \\ A := (\Phi_1, \dots, \Phi_p) & (d \times dp) \\ X_t^0 := \begin{pmatrix} X_t - \mu \\ \vdots \\ X_{t-p+1} - \mu \end{pmatrix} & (dp \times 1) \\ Y := (X_0^0, \dots, X_{T-1}^0) & (dp \times T) \\ x^0 := \text{vec}(X^0) & (dT \times 1) \\ \alpha := \text{vec}(A) & (d^2p \times 1) \end{cases}$$

On peut réécrire l'expression (2.24), pour $t = 1, \dots, T$ par :

$$X^0 = AY + E \quad (2.25)$$

Ce qui implique que

$$x^0 = (Y' \otimes I_d)\alpha + e \quad (2.26)$$

On en déduit facilement que l'estimateur des moindres carrés est donné par l'expression suivante :

$$\hat{\alpha} = ((YY')^{-1}Y \otimes I_d) x^0 \quad (2.27)$$

ou encore

$$\hat{A} = X^0 Y' (Y Y')^{-1} \quad (2.28)$$

De plus, si X_t est stable et ε_t un bruit blanc standard, on montre aisément que

$$\sqrt{T}(\hat{\alpha} - \alpha) \xrightarrow{\mathcal{L}} \mathcal{N}(0, \Gamma_{\hat{\alpha}}) \quad (2.29)$$

ou

$$\Gamma_{\hat{\alpha}} = \Gamma_X(0)^{-1} \otimes \Sigma_{\varepsilon} \quad (2.30)$$

et $\Gamma_X(0) := \mathbb{E}(X_t^0 X_t^{0'})$.

2.1.2.2 Estimation de la moyenne du processus

Très souvent, la moyenne μ du processus est inconnue. Dans ce cas, elle est estimée par le vecteur de moyenne empirique \bar{X} défini par

$$\bar{X} = \frac{1}{T} \sum_{t=1}^{t=T} X_t \quad (2.31)$$

En utilisant l'expression (2.24), \bar{X} peut être réécrit comme suit :

$$\begin{aligned} \bar{X} &= \mu + \Phi_1 \left[\bar{X} + \frac{1}{T}(X_0 - X_T) \right] + \dots \\ &+ \Phi_p \left[\bar{X} + \frac{1}{T}(X_{-p+1} + \dots + X_0 - X_{T-p+1} - \dots - X_T) - \mu \right] + \frac{1}{T} \sum_{t=1}^{t=T} \varepsilon_t \end{aligned}$$

Ainsi

$$(I_d - \Phi_1 - \dots - \Phi_p)(\bar{X} - \mu) = \frac{1}{T} z_T + \frac{1}{T} \sum_{t=1}^{t=T} \varepsilon_t \quad (2.32)$$

$$\text{avec } z_t = \sum_{j=1}^{j=p} \left[\sum_{i=0}^{i=j-1} (X_{0-i} - X_{T-i}) \right]$$

Propriété 2.3

z_t possède les propriétés suivantes :

1. $\mathbb{E} \left(\frac{z_t}{\sqrt{T}} \right) = \frac{1}{\sqrt{T}} \mathbb{E}(z_t) = 0$
2. $\text{Var} \left(\frac{z_t}{\sqrt{T}} \right) = \frac{1}{T} \text{Var}(z_t) \xrightarrow{T \rightarrow +\infty} 0$

parce que X_t est stable. En d'autres termes, z_t/\sqrt{T} converge vers zéro en moyenne quadratique. Il suit que $\sqrt{T} (I_d - \Phi_1 - \dots - \Phi_p) (\bar{X} - \mu)$ possède la même distribution asymptotique que $\sum_t \varepsilon_t / \sqrt{T}$. Par conséquent, en appliquant le théorème central limite, il s'en suit que

$$\frac{1}{\sqrt{T}} \sum_{t=1}^{t=T} \varepsilon_t \xrightarrow{\mathcal{L}} \mathcal{N}(0, \Sigma_\varepsilon) \quad (2.33)$$

Si ε_t est un bruit blanc standard, nous avons le résultat suivant :

Proposition 2.1 *Propriétés asymptotiques de la moyenne de l'échantillon*

Si le processus VAR(p) défini par l'expression (2.24) est stable et ε_t est un bruit blanc standard, alors

$$\sqrt{T}(\bar{X} - \mu) \xrightarrow{\mathcal{L}} \mathcal{N}(0, \Sigma_{\bar{X}}) \quad (2.34)$$

où $\Sigma_{\bar{X}} = (I_d - \Phi_1 - \dots - \Phi_p)^{-1} \Sigma_\varepsilon (I_d - \Phi_1 - \dots - \Phi_p)'^{-1}$. En particulier, $\bar{X} \xrightarrow{\mathbb{P}} \mu$.

Puisque $\mu = (I - \Phi_1 - \dots - \Phi_p)^{-1} c$, alors un estimateur alternatif pour le processus moyen est obtenu à partir de l'estimateur des moindres carrés généralisés et on a :

$$\hat{\mu} = (I_d - \hat{\Phi}_1 - \dots - \hat{\Phi}_p)^{-1} \hat{c} \quad (2.35)$$

L'estimateur $\hat{\mu}$ est consistant et possède une distribution asymptotiquement normale, c'est-à-dire

$$\sqrt{T}(\hat{\mu} - \mu) \xrightarrow{\mathcal{L}} \mathcal{N} \left(0, \frac{\partial \mu}{\partial \beta'} (\Gamma^{-1} \otimes \Sigma_\varepsilon) \frac{\partial \mu'}{\partial \beta} \right) \quad (2.36)$$

On peut montrer facilement que

$$\frac{\partial \mu}{\partial \beta'} (\Gamma^{-1} \otimes \Sigma_\varepsilon) \frac{\partial \mu'}{\partial \beta} = \Sigma_{\bar{X}} \quad (2.37)$$

Ainsi, les estimateurs $\hat{\mu}$ et \bar{X} sont asymptotiquement équivalents.

2.1.2.3 Estimation avec une moyenne de processus inconnue

Si la moyenne μ du processus est inconnue, elle est remplacée par \bar{X} dans le vecteur. L'estimateur des moindres carrés

$$\hat{\hat{\alpha}} = \left((\widehat{Y\hat{Y}'})^{-1} \widehat{Y} \otimes I_d \right) \hat{x}^0 \quad (2.38)$$

est asymptotiquement équivalent à $\hat{\alpha}$. Plus précisément, on peut aisément montrer que, sous certaines conditions de la proposition (2.1),

$$\sqrt{T} \left(\hat{\hat{\alpha}} - \alpha \right) \xrightarrow{\mathcal{L}} \mathcal{N} \left(0, \Gamma_X(0) \otimes \Sigma_\varepsilon \right) \quad (2.39)$$

où $\Gamma_X(0) := \mathbb{E} \left(X_t^0 X_t^{0'} \right)$.

2.1.3 Estimation par maximum de vraisemblance

Considérons le processus VAR(p) centré suivant :

$$X_t = \Phi_1 X_{t-1} + \Phi_2 X_{t-2} + \dots + \Phi_p X_{t-p} + \varepsilon_t \quad (2.40)$$

où ε_t est un bruit blanc de matrice de variance covariance Σ_ε .

On écrit la vraisemblance conditionnellement à toutes les valeurs passées du processus :

$$L(X_1, \dots, X_T) = \prod_{t=1}^T L(X_t | \underline{X}_{t-1}) \quad (2.41)$$

où \underline{X}_{t-1} désigne tout le passé de X_t jusqu'à la date $(t-1)$ incluse. La vraisemblance s'écrit alors :

$$L(X_1, \dots, X_T; \theta) = \sum_{t=1}^T \frac{1}{(2\pi)^{d/2} \sqrt{\det \Sigma_\varepsilon}} \exp \left[-\frac{1}{2} \sum_{t=1}^T \varepsilon_t' \Sigma_\varepsilon^{-1} \varepsilon_t \right] \quad (2.42)$$

On en déduit l'expression de la log-vraisemblance :

$$\mathcal{L}(X_1, \dots, X_T; \theta) = -\frac{dT}{2} \log(2\pi) - \frac{T}{2} \log(\det \Sigma_\varepsilon) - \frac{1}{2} \sum_{t=1}^T \varepsilon_t' \Sigma_\varepsilon^{-1} \varepsilon_t \quad (2.43)$$

On maximise ensuite cette expression afin d'obtenir les estimations de Φ_1, \dots, Φ_p et de Σ_ε , c'est - à - dire

$$\hat{\theta}_{MV} = \max_{\theta \in \mathbb{R}^{d \times dp}} \mathcal{L}(X_1, \dots, X_T; \theta) \quad (2.44)$$

2.1.4 Estimation par équation de Yule - Walker

L'estimateur des moindres carrés peut aussi être obtenu par les équations de Yule - Walker. Les équations de Yule - Walker d'un processus pour $h > 0$ s'écrivent :

$$\Gamma_X(h) = \Phi_1 \Gamma_X(h-1) + \Phi_2 \Gamma_X(h-2) + \dots + \Phi_p \Gamma_X(h-p) \quad (2.45)$$

Pour $h = 0$, on a

$$\Sigma_\varepsilon = \Gamma_X(0) - \sum_{j=1}^p \Phi_j \Gamma_X(j) \quad (2.46)$$

Comme dans le cas univarié, les matrices $\Gamma_X(0), \dots, \Gamma_X(p)$ peuvent être estimées à partir de la suite (X_1, \dots, X_T) . Ainsi, un estimateur naturel de la matrice de covariance $\Gamma_X(h) = \mathbb{E} [(X_{t+h} - \mu)(X_t - \mu)']$ est

$$\hat{\Gamma}_X(h) = \begin{cases} \frac{1}{T} \sum_{t=1}^{T-h} (X_{t+h} - \bar{X})(X_t - \bar{X})' & \text{pour } 0 \leq h \leq T-1 \\ \hat{\Gamma}_X'(-h) & \text{pour } -T+1 \leq h < 0 \end{cases} \quad (2.47)$$

Les estimateurs de Yule - Walker $\hat{\Phi}_1, \dots, \hat{\Phi}_p$ et $\hat{\Sigma}_\varepsilon$ sont obtenus en remplaçant $\Gamma_X(j)$ par son estimateur $\hat{\Gamma}_X(j)$ $j = 0, \dots, p$ et en résolvant les équations pour Φ_1, \dots, Φ_p et Σ_ε .

Pour des processus stables, l'estimateur de Yule - Walker possède les mêmes propriétés asymptotiques que celui des moindres carrés. Cependant, l'estimateur de Yule-Walker produit uniquement des estimations dans les régions stables. En d'autres termes, les processus estimé est toujours stable.

```
# ACF plot for sample data
results.plot_sample_acorr(nlags=7, linewidth=8);
plt.show()
```

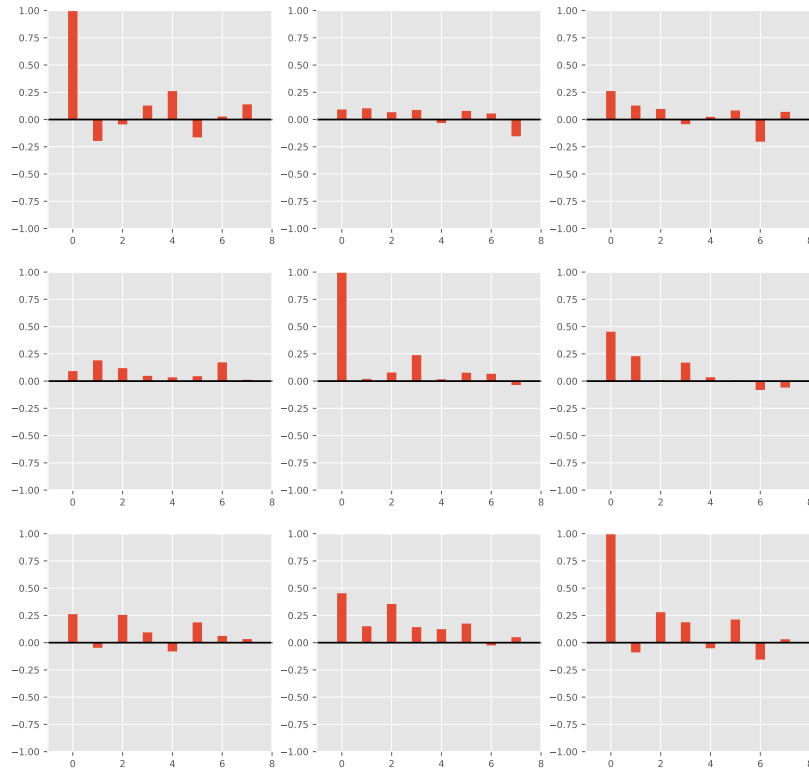


Figure 2.4 – Some estimated autocorrelations of investment/income/consumption system

2.2 Détermination du nombre de retards

Différentes approches sont proposées pour choisir le bon ordre du modèle. Les plus utilisées sont le test du rapport de vraisemblance et les critères d'information.

2.2.1 Test du rapport de maximum de vraisemblance

L'objectif de ce test est de comparer deux modèles, l'un qui est dit contraint et l'autre qui est le modèle complet (ou non contraint). Rappelons que la statistique du rapport de vraisemblance, notée λ_{RV} , est

$$\lambda_{RV} = 2 [\mathcal{L}^{nc} - \mathcal{L}^c] \quad (2.48)$$

où \mathcal{L}^{nc} (respectivement \mathcal{L}^c) représente la fonction de log - vraisemblance associé au modèle non contraint (respectivement contraint). Dans notre contexte, on teste les hypothèses suivantes :

$$H_0 : \Phi_{p+1} = 0 : \text{processus } VAR(p) \quad \text{contre} \quad H_1 : \Phi_{p+1} \neq 0 : \text{Processus } VAR(p+1) \quad (2.49)$$

La technique du test consiste à estimer un modèle contraint ($VAR(p)$) et un modèle non contraint ($VAR(p+1)$) et à effectuer le rapport des log - vraisemblances.

La log - vraisemblance d'un processus $VAR(p)$ est définie par l'équation (2.43). De plus, on a montré que $\sum_{t=1}^{t=T} \varepsilon_t' \Sigma_\varepsilon \varepsilon_t = dT$. Par conséquent, la log - vraisemblance estimée du modèle contraint s'écrit :

$$\mathcal{L}^c = -\frac{dT}{2} \log(2\pi) - \frac{T}{2} \log \det \tilde{\Sigma}_\varepsilon^c - \frac{1}{2} dT \quad (2.50)$$

et celle du modèle non contraint

$$\mathcal{L}^{nc} = -\frac{dT}{2} \log(2\pi) - \frac{T}{2} \log \det \tilde{\Sigma}_\varepsilon^{nc} - \frac{1}{2} dT \quad (2.51)$$

où $\tilde{\Sigma}_\varepsilon^c$ (respectivement $\tilde{\Sigma}_\varepsilon^{nc}$) désigne l'estimateur du maximum de vraisemblance de la matrice de variance covariance des résidus du modèle contraint (respectivement non contraint).

Ainsi, l'équation (2.48) se réécrit comme suit :

$$\lambda_{RV} = T \log \left(\frac{\det \tilde{\Sigma}_\varepsilon^c}{\det \tilde{\Sigma}_\varepsilon^{nc}} \right) \quad (2.52)$$

Sous l'hypothèse nulle, cette statistique suit une loi de χ^2 à d^2 degrés de liberté.

Si l'on accepte l'hypothèse nulle, on peut effectuer un deuxième test :

$$H_0 : \Phi_p = 0 : \text{processus } VAR(p-1) \quad \text{contre} \quad H_1 : \Phi_p \neq 0 : \text{Processus } VAR(p) \quad (2.53)$$

Ce test s'effectue de la même façon que précédemment. On a ainsi, une séquence de tests emboîtés dont le but est de déterminer l'ordre p du processus VAR. Ces tests sont conditionnels parce qu'ils dépendent des résultats des tests précédents. Le nombre de tests effectués au total est donc

fonction du nombre d'hypothèses nulles non rejetées. On définit de ce fait, un ordre p maximal qu'on note M .

Dans le cas d'un VAR(p), la statistique du rapport de vraisemblance pour la i -ième hypothèse nulle est notée $\lambda_{RV}(i)$ et est donnée par :

$$\lambda_{RV}(i) = T \log \left(\frac{\det \tilde{\Sigma}_\varepsilon(M-i)}{\det \tilde{\Sigma}_\varepsilon(M-i+1)} \right) \quad (2.54)$$

La statistique $\lambda_{RV}(i)$ suit une loi χ^2 à d^2 degrés de liberté sous $H_0^i : \Phi_{M-i+1} = 0$. En effet, le nombre de contraintes équivaut au nombre d'éléments de la dernière matrice de coefficients, c'est-à-dire d^2 car elle est une matrice carré d'ordre d .

Proposition 2.2 *Test de rapport de vraisemblance à partir de la statistique de Fisher*

De façon alternative, la statistique de Fisher définie par

$$\lambda_{RV}^F(i) = \lambda_{RV}(i)/d^2 \quad (2.55)$$

suit sous l'hypothèse nulle une loi de Fisher à d^2 et $(T - d(M - i + 1) - 1)$ degrés de liberté.

Remarque 2.1

Les modèles choisis grâce à cette méthode ont tendance à être moins parcimonieux que ceux sélectionnés par les critères d'information.

2.2.2 Les critères d'information

Dans le processus de sélection de modèle, diverses méthodes parmi lesquelles figurent les critères d'information sont mises à la disposition de l'analyste. Il est important de tenir compte de l'objectif que l'on cherche à atteindre lorsqu'on sélectionne l'ordre du modèle VAR(p). Nous allons présenter 4 critères qui sont à minimiser. L'ordre choisi pour un critère est celui du modèle VAR(p) qui a la plus petite valeur du critère parmi les modèles candidats. On définit M l'ordre maximum que notre modèle ajusté devrait avoir.

2.2.2.1 L'erreur limite de prédiction ou FPE (Final Predictor Error)

Ce critère a été proposé par Akaike (1969). Il propose de baser le choix du modèle sur la minimisation de l'erreur quadratique moyenne prédite à l'horizon 1. On a :

$$FPE(p) = \det \left[\frac{T+dp+1}{T} \frac{T}{T-dp-1} \tilde{\Sigma}_\varepsilon(p) \right] = \left| \frac{T+dp+1}{T-dp-1} \right|^d \det \tilde{\Sigma}_\varepsilon(p) \quad (2.56)$$

où $\tilde{\Sigma}_\varepsilon(p)$ est l'estimateur du maximum de vraisemblance de Σ_ε lorsqu'un ordre égal à p est choisi, c'est-à-dire si on a ajusté un VAR(p). L'ordre $\hat{p}(FPE)$ et la valeur de l'ordre du modèle dont le critère FPE est le plus petit parmi les candidats ($p = 0, 1, \dots, M$).

2.2.2.2 Le critère d'information d'Akaike ou AIC

Il a été proposé par Akaike (1973) (Akaike (1974)) et

$$AIC(p) = \ln \det \tilde{\Sigma}_\varepsilon(p) + \frac{2pd^2}{T} \quad (2.57)$$

pour $(p = 0, 1, \dots, M)$. On choisit l'ordre $\hat{p}(AIC)$ du modèle dont le critère AIC est minimisé.

2.2.2.3 Le critère d'information Bayésien ou BIC

Il a été proposé par Schwarz (1978) et s'écrit :

$$BIC(p) = \ln \det \tilde{\Sigma}_\varepsilon(p) + \frac{pd^2}{T} \ln(T) \quad (2.58)$$

pour $(p = 0, 1, \dots, M)$. On choisit l'ordre $\hat{p}(BIC)$ du modèle dont le critère BIC est *minimisé*.

2.2.2.4 Le critère d'information de Hannan-Quinn ou HQIC

Il a été proposé par Hannan and Quinn (1979) et Quinn (1980). Sa formule est :

$$HQIC(p) = \ln \det \tilde{\Sigma}_\varepsilon(p) + \frac{2pd^2}{T} \ln(\ln(T)) \quad (2.59)$$

pour $(p = 0, 1, \dots, M)$. On choisit l'ordre $\hat{p}(HQ)$ du modèle dont le critère BIC est minimisé.

Propriété 2.4 Comparaison des critères AIC, SC et HQIC sur des petits échantillons

Soit $X_{-M+1}, \dots, X_0, X_1, \dots, X_T$ une série temporelle d -dimensionnel et supposons que les modèles $VAR(p), p = 1, \dots, M$ sont ajustés à $X_1, X \dots, X_T$. Alors es relations suivantes sont satisfaites :

$$\begin{cases} \hat{p}(SC) \leq \hat{p}(AIC) & \text{si } T \geq 8 \\ \hat{p}(SC) \leq \hat{p}(HQIC) & \forall T \\ \hat{p}(HQIC) \leq \hat{p}(AIC) & \text{si } T \geq 16 \end{cases} \quad (2.60)$$

Ces critères peuvent être comparés selon différentes propriétés. En effet, la notion de performance d'un critère dépend de la définition que l'on donne à cette performance ou optimalité. Tout d'abord, lorsque l'objectif de l'analyse est la prédiction, on choisira le FPE ou l'AIC car ils préservent l'efficacité asymptotique, ce que BIC et HQIC ne font pas. On a que $\ln FPE(p) = AIC(p) + 2d/T + o(T^{-2})$, ce qui permet d'affirmer que ces deux critères ont tendance à sélectionner le même modèle pour des échantillons plus ou moins larges puisque la valeur $2d/T$ ne dépend pas de l'ordre m .

Exemple 2.2 Choix de p à l'aide des critères d'information

Pour illustrer ces procédures de sélection de l'ordre d'un processus VAR, nous considérons notre base de données composées de invest, income et cons. Nous utilisons la méthode `model.select_order()` du modèle construit en spécifiant le retard maximal.

```
# Choix de p
print(model.select_order(4).summary())

## VAR Order Selection (* highlights the minimums)
## =====
##      AIC      BIC      FPE      HQIC
## -----
## 0      -24.34    -24.24*  2.691e-11  -24.30*
## 1      -24.41    -24.03  2.500e-11  -24.26
## 2      -24.51*   -23.84  2.272e-11*  -24.24
## 3      -24.32    -23.37  2.748e-11  -23.94
## 4      -24.27    -23.03  2.910e-11  -23.78
## -----
```

Les critères de FPE et AIC font le choix d'un retard minimum égal à 2, $\hat{p}(FPE) = \hat{p}(AIC) = 2$. Nos résultats concordent à ceux fournis par la fonction `VARselect` du package R `vars` mais différent de ceux du tableau 4.5 du livre de Lütkepohl (2005).

Un constat que nous avons fait lorsque nous recherchions nous même la valeur p minimale à l'aide d'une boucle itérative est que les critères d'information pour différents modèles entraînés étaient différents de ceux trouver par la méthode `model.select_order()`. Nous avons obtenu les résultats suivants :

```
# Choix de p
import pandas as pd
select_p = pd.DataFrame(columns = ["AIC(p)", "BIC(p)", "FPE(p)", "HQIC(p)"],
                        index = [x for x in range(5)])

for p in range(5):
    res = VAR(DTrain).fit(p)
    select_p.iloc[p, :] = list([res.aic, res.bic, res.fpe, res.hqic])
select_p.insert(0, "p", list([x for x in range(5)]))
print(select_p)
```

```
##      p      AIC(p)      BIC(p) FPE(p)      HQIC(p)
## 0  0 -24.355407 -24.262707    0.0 -24.318393
## 1  1 -24.412376 -24.038744    0.0 -24.26333
## 2  2 -24.549441 -23.890541    0.0 -24.286858
## 3  3 -24.345218 -23.396607    0.0 -23.967573
## 4  4 -24.272974 -23.030094    0.0 -23.77872
```

Nous aboutissons à la même conclusion pour le choix de p .

2.3 Validation de modèle et prévision

Après avoir sélectionné le meilleur ordre de modèle et procédé à des estimations des paramètres associés à celui-ci, nous allons le valider. Il faudrait qu'il respecte les hypothèses sous-jacentes au modèle. Dans cette section, nous allons présenter les tests sur les résidus du modèle afin de vérifier s'ils sont bel et bien des bruits blancs.

2.3.1 Tests de bruit blanc des erreurs

Pour tester le manque de corrélation dans la série des résidus d'un modèle VAR(p), plusieurs tests sont généralement appliqués.

```
# Autocorrélogramme des résidus
results.plot_acorr(nlags=12, resid=True);
plt.show()
```

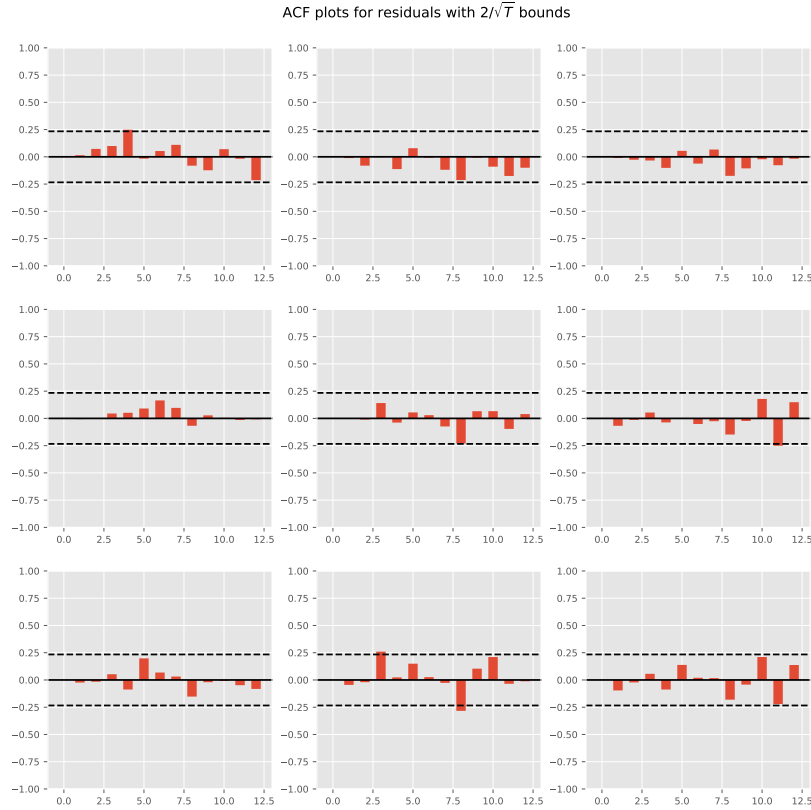


Figure 2.5 – Estimated residual autocorrelations with two-standard error bounds for the investment/income/consumption VAR(2) model.

Il y a des autocorrélations des résidus significatifs au retard 3, 4, 8 et 11.

2.3.1.1 Le test de Portmanteau

On veut vérifier si les erreurs de prévision $\hat{\varepsilon}_t$ résultant de l'estimation de notre modèle sont des réalisations d'un bruit blanc vectoriel. Comme dans le cas univarié, des tests Portmanteau multivariés existent. Nous allons présenter la statistique de test de Portmanteau modifiée proposée par Hosking (1980).

Supposons que le modèle a été estimé par la méthode des moindres carrés et est stable. L'hypothèse nulle est l'absence d'autocorrélation jusqu'à un certain retard « h » des corrélations croisées. On teste :

$$H_0 : R_h = (R_\varepsilon(1), \dots, R_\varepsilon(h)) = 0 \quad \text{contre} \quad H_1 : R_h \neq 0 \quad (2.61)$$

où $R_\varepsilon(k), k = 0, \dots, h$ représente la matrice d'autocorrélation de ε_t . On a :

$$R_\varepsilon(k) = D^{-1}\Gamma_\varepsilon(k)D^{-1} \quad (2.62)$$

où D est la matrice diagonale composée des racines carrées des éléments de la diagonale de $\Gamma_\varepsilon(0)$ où $\hat{\Gamma}_X(i)$ est l'estimateur de la matrice d'autocovariance de ε de retard i et

$$\hat{\Gamma}_\varepsilon(i) = \frac{1}{T} \sum_{t=i+1}^{i=T} \hat{\varepsilon}_t \hat{\varepsilon}'_{t-i}, \quad i = 0, \dots, h < T \quad (2.63)$$

Définition 2.1 *Statistique de Box - Pierce*

La statistique du test de Portmanteau de Box and Pierce (1970) est définie comme suit :

$$\begin{aligned} Q_h &= T \sum_{i=1}^{i=T} \text{trace} \left(\hat{R}'_\varepsilon(i) \hat{R}_\varepsilon^{-1}(0) \hat{R}_\varepsilon(i) \hat{R}_\varepsilon^{-1}(0) \right) \\ &= T \sum_{i=1}^{i=T} \text{trace} \left(\hat{R}'_\varepsilon(i) \hat{R}_\varepsilon^{-1}(0) \hat{R}_\varepsilon(i) \hat{R}_\varepsilon^{-1}(0) \hat{D}^{-1} \hat{D} \right) \\ &= T \sum_{i=1}^{i=T} \text{trace} \left(\hat{R}'_\varepsilon(i) \hat{R}_\varepsilon^{-1}(0) \hat{R}_\varepsilon(i) \hat{R}_\varepsilon^{-1}(0) \right) \\ &= T \sum_{i=1}^{i=T} \text{trace} \left(\hat{D} \hat{R}'_\varepsilon(i) \hat{D} \hat{D}^{-1} \hat{R}_\varepsilon^{-1}(0) \hat{D}^{-1} \hat{D} \hat{R}_\varepsilon(i) \hat{D} \hat{D}^{-1} \hat{R}_\varepsilon^{-1}(0) \hat{D}^{-1} \right) \\ &= T \sum_{i=1}^{i=T} \text{trace} \left(\hat{\Gamma}'_\varepsilon(i) \hat{\Gamma}_\varepsilon^{-1}(0) \hat{\Gamma}_\varepsilon(i) \hat{\Gamma}_\varepsilon^{-1}(0) \right) \end{aligned}$$

Sous H_0 , cette statistique suit une loi de χ^2 à $d^2(h-p)$ degrés de liberté.

Définition 2.2 *Statistique de Ljung - Box*

Pour un échantillon de petite taille, Ljung and Box (1978) propose la statistique de test de Portmanteau multivariée modifiée, notée \bar{Q}_h et définie par :

$$\begin{aligned} \bar{Q}_h &= T^2 \sum_{i=1}^{i=T} (T-i)^{-1} \text{trace} \left(\hat{R}'_\varepsilon(i) \hat{R}_\varepsilon^{-1}(0) \hat{R}_\varepsilon(i) \hat{R}_\varepsilon^{-1}(0) \right) \\ &= T^2 \sum_{i=1}^{i=T} (T-i)^{-1} \text{trace} \left(\hat{R}'_\varepsilon(i) \hat{R}_\varepsilon^{-1}(0) \hat{R}_\varepsilon(i) \hat{R}_\varepsilon^{-1}(0) \hat{D}^{-1} \hat{D} \right) \\ &= T^2 \sum_{i=1}^{i=T} (T-i)^{-1} \text{trace} \left(\hat{R}'_\varepsilon(i) \hat{R}_\varepsilon^{-1}(0) \hat{R}_\varepsilon(i) \hat{R}_\varepsilon^{-1}(0) \right) \\ &= T^2 \sum_{i=1}^{i=T} (T-i)^{-1} \text{trace} \left(\hat{D} \hat{R}'_\varepsilon(i) \hat{D} \hat{D}^{-1} \hat{R}_\varepsilon^{-1}(0) \hat{D}^{-1} \hat{D} \hat{R}_\varepsilon(i) \hat{D} \hat{D}^{-1} \hat{R}_\varepsilon^{-1}(0) \hat{D}^{-1} \right) \\ &= T^2 \sum_{i=1}^{i=T} (T-i)^{-1} \text{trace} \left(\hat{\Gamma}'_\varepsilon(i) \hat{\Gamma}_\varepsilon^{-1}(0) \hat{\Gamma}_\varepsilon(i) \hat{\Gamma}_\varepsilon^{-1}(0) \right) \end{aligned}$$

Si H_0 est vraie, la statistique \bar{Q}_h suit une loi de χ^2 à $d^2(h-p)$.

Cette statistique est une meilleure alternative à $Q_h = T \sum_{i=1}^{i=h} \text{tr} \left(\hat{\Gamma}'_{\varepsilon}(i) \hat{\Gamma}_{\varepsilon}^{-1}(0) \hat{\Gamma}_{\varepsilon}(i) \hat{\Gamma}_{\varepsilon}^{-1}(0) \right)$ même si elles sont asymptotiquement équivalentes¹ parce que certains chercheurs (Davies, Triggs, and Newbold (1977), Ljung and Box (1978), Hosking (1980)) ont constaté que la puissance du test relié à Q_h est faible dans le cas des petits échantillons.

Exemple 2.3 *Test de Portmanteau sur les résidus du modèle VAR(2)*

La méthode `res.test_whiteness` effectue le test de Portmanteau multivarié. Pour notre exemple avec les données macroéconomiques, on a le résultat du test de Portmanteau suivant :

```
# Extraction des résultats du test
def whiteness_output(res):
    print("=="*35)
    print(res.title)
    print(res.signif_str)
    print("=="*35)
    print(res.h0)
    print("=="*35)
    df = pd.DataFrame({
        "Test statistic": res.test_statistic, "Critical value" : res.crit_value,
        "df" : res.df, "p-value" : res.pvalue, "signif" : res.signif},
        index = ["Portmanteau test"])
    print(df)
    print("=="*35)
    print(res.conclusion_str)
    print("=="*35)
# Test d'autocorrélation (Portmanteau test)
white_test = results.test_whiteness(nlags=12, signif=0.05, adjusted=True)
whiteness_output(white_test)

## =====
## Adjusted Portmanteau-test for residual autocorrelation
## at 5% significance level
## =====
## H_0: residual autocorrelation up to lag 12 is zero
## =====
##              Test statistic  Critical value  df   p-value  signif
## Portmanteau test           81.933526       113.14527  90  0.715697   0.05
## =====
## Conclusion: fail to reject H_0
## =====
```

On ne peut pas rejeter l'hypothèse d'un bruit blanc pour les résidus estimés. au seuil de 5%.

Nous effectuons le test de portmanteau des retards h allant de 3 à 12.

1. Notons que $T / [T^2(T-i)^{-1}] \xrightarrow{T \rightarrow +\infty} 1$.

```
# Test de portmanteau pour h = 3,...,12 au seuil de 5%
pm_var = pd.DataFrame(columns = [f"Q({x+1})" for x in range(2,12)],
                      index = ["test statistic", "df", "Critical value", "p-value"])
for i in range(3,13):
    res = results.test_whiteness(nlags=i, signif=0.05, adjusted=True)
    pm_var.iloc[:,i-3] = [res.test_statistic, res.df, res.crit_value, res.pvalue]
```

Table 2.2 – Test de portmanteau pour h allant de 3 à 12

	Q(3)	Q(4)	Q(5)	Q(6)	Q(7)	Q(8)	Q(9)	Q(10)	Q(11)	Q(12)
test statistic	9.69	22.07	27.82	32.59	37.91	48.31	52.23	62.12	73.05	81.93
df	9.00	18.00	27.00	36.00	45.00	54.00	63.00	72.00	81.00	90.00
Critical value	16.92	28.87	40.11	51.00	61.66	72.15	82.53	92.81	103.01	113.15
p-value	0.38	0.23	0.42	0.63	0.76	0.69	0.83	0.79	0.72	0.72

Au seuil de 5%, on ne peut pas rejeter l'hypothèse nulle d'un bruit blanc sur les résidus estimés pour les retards allant de 3 à 12.

2.3.1.2 Test du multiplicateur de Lagrange

Une autre manière de tester l'autocorrélation des résidus est de considérer que les erreurs du modèle VAR suivent VAR(h), c'est - à - dire que :

$$\varepsilon_t = \Pi_1 \varepsilon_{t-1} + \dots + \Pi_h \varepsilon_{t-h} + \mu_t \quad (2.64)$$

où μ_t est un bruit blanc multivarié. Ainsi, on a les hypothèses de test suivantes :

$$H_0 : \Pi_1 = \dots = \Pi_h = 0 \quad \text{contre} \quad H_1 : \exists j \in \{1, \dots, h\} / \Pi_j \neq 0 \quad (2.65)$$

Breusch - Godfrey (Breusch (1978), L. G. Godfrey (1978), L. Godfrey (1988)) propose une statistique LM du test basée sur la régression auxiliaire suivante :

$$\hat{\varepsilon}_t = m + A_1 X_{t-1} + \dots + A_p X_{t-p} + \Pi_1 \hat{\varepsilon}_{t-1} + \dots + \Pi_h \hat{\varepsilon}_{t-h} + \eta_t \quad (2.66)$$

La statistique de test est définie par :

$$LM_h = T(d - \text{trace}(\tilde{\Sigma}_\eta^{-1} \tilde{\Sigma}_\varepsilon)) \quad (2.67)$$

où $\tilde{\Sigma}_\eta^{-1}$ représente l'inverse de la matrice de variance covariance des résidus du modèle auxiliaire. Sous H_0 , cette statistique suit une loi de χ^2 à hd^2 degrés de liberté.

Remarque 2.2

Edgerton and Shukur (1999) ont proposé une correction du test qui tient compte des échantillons de petits tailles, elle est donnée par la statistique suivante :

$$F_{Rao}(h) = \left[\left(\frac{\det(\tilde{\Sigma}_\varepsilon)}{\det(\tilde{\Sigma}_\eta)} \right)^{1/s} - 1 \right] \frac{Ns - \frac{1}{2}d^2h + 1}{d^2h} \quad (2.68)$$

avec $s = \sqrt{\frac{(dh-2)(dh+2)}{d^2(1+h^2)-5}}$ et $N = T - dp - 1 - dh - \frac{1}{2}(d - dh + 1)$.

Cette statistique suit sous H_0 une loi de Fisher à hd^2 et $Ns - \frac{1}{2}hd^2 + 1$ degrés de liberté.

Contrairement au test de Portmanteau qui doit être utilisé pour des valeurs h raisonnables, le test du multiplicateur de Lagrange est plus approprié pour des échantillons de petite taille.

2.3.2 Test d'hétéroscédasticité

Dans la fonction `arch.test` du package `vars` de R, le test du multiplicateur de Lagrange (ARCH - LM Test), dans le cas multivarié, est défini comme suit :

$$\text{vech}(\hat{\varepsilon}_t \hat{\varepsilon}_t') = B_0 + B_1 \text{vech}(\hat{\varepsilon}_{t-1} \hat{\varepsilon}_{t-1}') + \dots + B_q \text{vech}(\hat{\varepsilon}_{t-q} \hat{\varepsilon}_{t-q}') + v_t \quad (2.69)$$

où v_t est le terme d'erreur. B_0 est un vecteur de dimension $\left(\frac{1}{2}d(d+1) \times 1\right)$ et les matrices de coefficients $B_i, i = 1, \dots, q$ sont de format $\left(\frac{1}{2}d(d+1) \times \frac{1}{2}d(d+1)\right)$. On teste les hypothèses suivantes :

$$H_0 : B_1 = \dots = B_q \quad \text{contre} \quad H_1 : \exists i \in \{1, \dots, q\} / B_i \neq 0 \quad (2.70)$$

La statistique de test définie par

$$VARCH_{LM}(q) = \frac{T}{2}d(d+1)R_m^2 \quad (2.71)$$

avec

$$R_m^2 = 1 - \frac{2}{d(d+1)} \text{tr}(\widehat{\Sigma} \widehat{\Sigma}_0^{-1}) \quad (2.72)$$

où $\widehat{\Sigma}$ est la matrice de variance covariance de l'erreur du modèle précédent. Sous l'hypothèse nulle, cette statistique suit une loi de χ^2 à $qd^2(d+1)/4$.

2.3.3 Test de normalité

Pour étudier la normalité des résidus, on peut se servir des tests de normalité univariés qui vont porter successivement sur chacune des séries résiduelles des d equations. On peut citer le test de Jarque - Bera de normalité des résidus.

Proposition 2.3 *Statistique de Jarque - Bera univariée*

Ce test est basé sur le coefficient d'aplatissement (kurtosis, K) et le coefficient d'asymétrie (skewness, S) définis comme suit (dans le cas univarié) :

$$S = \frac{\mu_3}{\mu_2^{3/2}} \quad \text{et} \quad K = \frac{\mu_4}{\mu_2^2} \quad (2.73)$$

L'idée est de vérifier si ces coefficients sont égaux à ceux d'une variable qui suit une loi normale, c'est-à-dire, respectivement 3 et 0. On teste alors

$$\begin{cases} H_0 : S = 0 \quad \text{et} \quad K = 0 \\ H_1 : S \neq 0 \quad \text{ou} \quad K \neq 3 \end{cases} \quad (2.74)$$

Sous l'hypothèse de normalité

$$\sqrt{T/6}\widehat{S} \sim \mathcal{N}(0,1) \quad \text{et} \quad \sqrt{T/24}(\widehat{K} - 3) \sim \mathcal{N}(0,1) \quad (2.75)$$

et \widehat{S} et \widehat{K} sont indépendants et obtenus en prenant les estimateurs empiriques des moments dans les définitions précédentes. Jarque and Bera (1987) ont montré que

$$JB = \frac{T}{6} \left(\widehat{S}^2 + \frac{\widehat{K} - 3}{4} \right) \sim \chi_2^2 \quad (2.76)$$

où T est la taille de l'échantillon.

On rejette l'hypothèse nulle pour des valeurs élevées de JB ou encore lorsque les valeurs p sont inférieures au seuil de significativité du test.

Exemple 2.4 Test de normalité de Jarque et Bera univarié

Nous appliquons le test de normalité univarié sur les résidus estimés de notre modèle VAR(2) estimé. Tout d'abord, affichons l'histogramme des résidus :

```
# Histogramme et densité des résidus
import scipy.stats as st
import seaborn as sns
fig, axes = plt.subplots(1,3,figsize=(16,6))
for idx,name in enumerate(results.resid.columns):
    sns.histplot(data=results.resid[name],binwidth=0.01,kde=True,ax=axes[idx],
                 label="Kernel density",color = "black",stat="proportion");
    axes[idx].set_xlabel("value");
    axes[idx].set_ylabel(f"resid {name} Density");
    axes[idx].legend();
plt.tight_layout();
plt.show()

# Test de Jarque - Bera univarié
from statsmodels.stats.stattools import jarque_bera
jb_resid = pd.DataFrame(columns = ["statistic","pvalue","skewness","kurtosis"],
                        index = results.resid.columns)
for name in results.resid.columns:
    res = jarque_bera(results.resid[[name]])
    jb_resid.loc[name,:] = list([res[0],res[1],res[2],res[3]])
```

Nos pvalues sont inférieures au seuil de 5%, on rejette l'hypothèse de normalité des résidus.

Proposition 2.4 Statistique de Jarque - Bera multivariée

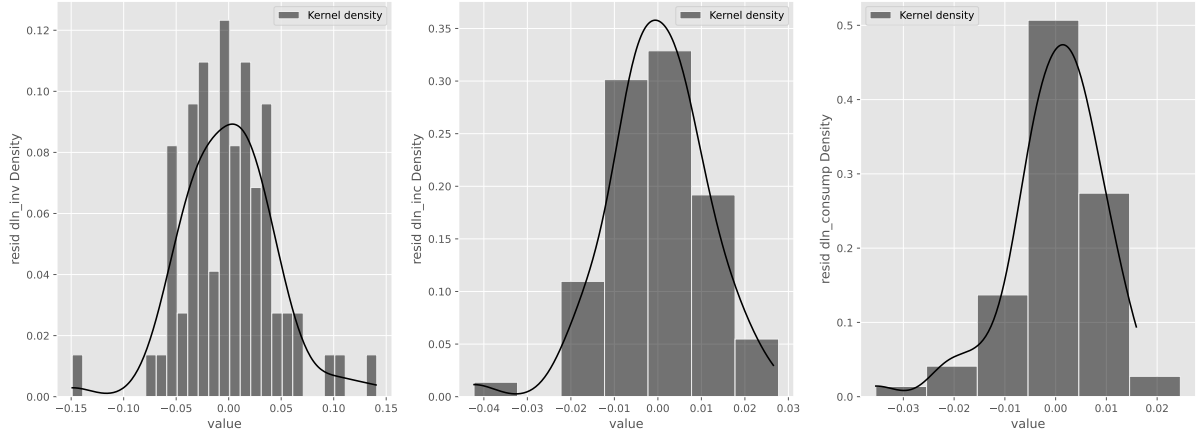


Figure 2.6 – Histogramme et densité des résidus

Table 2.3 – Test de normalité de Jarque - Bera univarié

	statistic	pvalue	skewness	kurtosis
dln_inv	10.21696	0.00604526	0.1388345	4.811602
dln_inc	11.98427	0.002498321	-0.5063547	4.707177
dln_consump	34.2508	3.652027e-08	-1.151224	5.441162

Soit ε_t un bruit blanc gaussien multivarié de dimension d , c'est - à - dire $\varepsilon_t \sim \mathcal{N}(\mu_\varepsilon, \Sigma_\varepsilon)$ et P , est la matrice triangulaire inférieure avec diagonale positive telle que $PP' = \Sigma_\varepsilon$; i.e. la décomposition de Cholesky de la matrice de variance et covariance. Alors :

$$w_t = (w_{1,t}, \dots, w_{d,t}) = P^{-1}(\varepsilon_t - \mu_\varepsilon) \sim \mathcal{N}(0, I_d) \quad (2.77)$$

En d'autres termes, les éléments de w_t sont indépendamment et identiquement distribués selon une loi normale. De plus,

$$\mathbb{E} \begin{bmatrix} w_{1,t}^3 \\ \vdots \\ w_{d,t}^3 \end{bmatrix} = 0 \quad \text{et} \quad \mathbb{E} \begin{bmatrix} w_{1,t}^4 \\ \vdots \\ w_{d,t}^4 \end{bmatrix} = 3_d$$

Pour construire la version multivariée de la statistique du test de Jarque - Bera, on considère les quantités suivantes :

$$\bar{\varepsilon} = \frac{1}{T} \sum_{t=1}^{t=T} \varepsilon_t \quad \text{et} \quad S_\varepsilon = \frac{1}{T-1} \sum_{t=1}^{t=T} (\varepsilon_t - \bar{\varepsilon})(\varepsilon_t - \bar{\varepsilon})'$$

et \tilde{P} la matrice définie par $\tilde{P}\tilde{P}' = S_\varepsilon$.²

De plus,

$$v_t = (v_{1,t}, \dots, v_{d,t}) = \tilde{P}^{-1}(\varepsilon_t - \bar{\varepsilon}), \quad t = 1, \dots, T \quad (2.78)$$

2. On a $\tilde{P} \xrightarrow{\mathbb{P}} P$ quand $T \rightarrow +\infty$.

$$b_1 = (b_{1,1}, \dots, b_{d,1}) \quad \text{avec} \quad b_{j,1} = \frac{1}{T} \sum_{t=1}^{t=T} v_{j,t}^3, \quad j = 1, \dots, d \quad (2.79)$$

et

$$b_2 = (b_{1,2}, \dots, b_{d,2}) \quad \text{avec} \quad b_{j,2} = \frac{1}{T} \sum_{t=1}^{t=T} v_{j,t}^4, \quad j = 1, \dots, d \quad (2.80)$$

Proposition 2.5 *Distribution asymptotique du skewness et du kurtosis*

Si ε_t est un bruit blanc gaussien de matrice de variance covariance Σ_ε non singulière et une moyenne μ_ε , c'est - à - dire $\varepsilon_t \sim \mathcal{N}(\mu_\varepsilon, \Sigma_\varepsilon)$, alors

$$\sqrt{T} \begin{bmatrix} b_1 \\ b_2 - 3_d \end{bmatrix} \sim \mathcal{N} \left(0, \begin{bmatrix} 6I_d & 0 \\ 0 & 24I_d \end{bmatrix} \right) \quad (2.81)$$

En d'autres termes, b_1 et b_2 sont asymptotiquement indépendants et identiquement distribués selon une loi normale. Ceci implique que

$$\lambda_S = \frac{Tb_1' b_1}{6} \sim \chi^2(d) \quad (2.82)$$

et

$$\lambda_K = \frac{T(b_2 - 3_d)'(b_2 - 3_d)}{24} \sim \chi^2(d) \quad (2.83)$$

La statistique du test de Jarque and Bera (1987) multivariée est définie par :

$$JB_{MV} = \lambda_S + \lambda_K \quad (2.84)$$

Sous H_0 , la statistique du test JM_{MV} est distribuée suivant la loi χ^2 à $2d$ degrés de liberté.

Proposition 2.6 *Distribution asymptotique du skewness et du kurtosis résiduels*

Soit $(X_t)_{t \in \mathbb{Z}}$, un processus VAR(p) de dimension d , stationnaire et stable où ε_t est un bruit blanc gaussien multivarié de moyenne nulle et de matrice de variance covariance Σ_ε non singulière. Soit $\hat{\Phi}_1, \dots, \hat{\Phi}_p$ les estimateurs consistents et asymptotiquement distribués selon une loi normale. Disposant des valeurs estimées des résidus, $\hat{\varepsilon}_t$, on définit :

$$\begin{cases} \hat{\varepsilon}_t &= (X_t - \bar{X}) - \hat{\Phi}_1(X_{t-1} - \bar{X}) - \dots - (X_{t-p} - \bar{X}), \quad t = 1, \dots, T \\ \hat{\Sigma}_\varepsilon &= \frac{1}{T - dp - 1} \sum_{t=1}^{t=T} \hat{\varepsilon}_t \hat{\varepsilon}_t' \end{cases} \quad (2.85)$$

et soit \hat{P} la matrice qui satisfait $\hat{P}\hat{P}' = \hat{\Sigma}_\varepsilon$ telle que $\hat{P} \xrightarrow{\mathbb{P}} P$ quand $T \rightarrow +\infty$. Deplus, on définit :

$$\hat{w}_t = (\hat{w}_{1,t}, \dots, \hat{w}_{d,t}) = \hat{P}' \hat{\varepsilon}_t \quad (2.86)$$

$$\hat{b}_1 = (\hat{b}_{1,1}, \dots, \hat{b}_{d,1}), \quad \text{avec} \quad \hat{b}_{j,1} = \frac{1}{T} \sum_{t=1}^{t=T} \hat{w}_{j,t}^3, \quad t = 1, \dots, T \quad (2.87)$$

et

$$\hat{b}_2 = (\hat{b}_{1,2}, \dots, \hat{b}_{d,2}), \quad \text{avec} \quad \hat{b}_{j,2} = \frac{1}{T} \sum_{t=1}^{t=T} \hat{w}_{j,t}^4, \quad t = 1, \dots, T \quad (2.88)$$

alors

$$\sqrt{T} \begin{bmatrix} \hat{b}_1 \\ \hat{b}_2 - 3_d \end{bmatrix} \sim \mathcal{N} \left(0, \begin{bmatrix} 6I_d & 0 \\ 0 & 24I_d \end{bmatrix} \right) \quad (2.89)$$

La proposition (2.6) implique que :

$$\hat{\lambda}_S = \frac{T \hat{b}_1' \hat{b}_1}{6} \sim \chi^2(d) \quad (2.90)$$

et

$$\frac{T(\hat{b}_2 - 3_d)'(\hat{b}_2 - 3_d)}{24} \sim \chi^2(d) \quad (2.91)$$

La statistique de Jarque and Bera (1987) multivariée s'écrit :

$$\widehat{JB}_{MV} = \hat{\lambda}_S + \hat{\lambda}_K \quad (2.92)$$

Sous l'hypothèse nulle de normalité, cette statistique suit une loi de χ^2 à $2d$ degrés de liberté.

Exemple 2.5 *Test de normalité des résidus du modèle VAR(2)*

La méthode `res.test_normality()` teste la normalité des résidus du modèle VAR à l'aide du test multivarié de Jarque - Bera.

```
# Extraction des résultats du test causalité
def normality_output(res):
    title = res.title + res.signif_str
    print("====*24)
    print(title)
    print("====*24)
    print(res.h0)
    print("====*24)
    df = pd.DataFrame({
        "Test statistic": res.test_statistic, "Critical value" : res.crit_value,
        "df" : res.df, "p-value" : res.pvalue, "signif" : res.signif},
        index = ["Jarque - Bera test"])
    print(df)
    print("====*24)
    print(res.conclusion_str)
```

```

print("==="*24)
# Test de normalité
norm_test = results.test_normality(signif=0.05)
normality_output(norm_test)

## =====
## normality (skew and kurtosis) test at 5% significance level
## =====
## H_0: data generated by normally-distributed process
## =====
##
##          Test statistic   Critical value   df   p-value   signif
## Jarque - Bera test      21.963595       12.591587   6   0.001229   0.05
## =====
## Conclusion: reject H_0
## =====

```

On a une p-value inférieure au seuil de 5%, on rejette l'hypothèse nulle de normalité des résidus.

2.3.4 Prévision

La prévision est l'un des principaux objectifs de l'analyse des séries temporelles multivariées. En pratique, nous sommes souvent confrontés à des situations où, à une période t particulière, il est nécessaire pour faire des déclarations sur les valeurs futures des variables $X_{1,t}, \dots, X_{d,t}$. Pour ce faire, nous avons un modèle pour le processus de génération de données et un ensemble d'informations, disons I_t , contenant les informations disponibles à la période t . Le processus de génération des données peut, par exemple, être un processus $VAR(p)$ et I_t peut contenir les variables passées et présentes du système considéré, c'est-à-dire $I_t = \{X_s | s \leq t\}$ où $X_s = (X_{1,s}, \dots, X_{d,s})'$. La période t est l'origine de la prévision et le nombre de période h dans le futur pour laquelle une prévision est souhaitée est l'horizon de prévision. L'optimalité des prévisions dépend d'une fonction de coût. Une prévision sera optimale si elle minimise le coût.

Considérons un processus $VAR(p)$:

$$X_t = \hat{c} + \hat{\Phi}_1 X_{t-1} + \dots + \hat{\Phi}_p X_{t-p} + \varepsilon_t \quad (2.93)$$

On suppose que p a été choisi, que \hat{c} et les $\hat{\Phi}_j$ ont été estimés et que la matrice de variance covariance associée à ε_t a été estimée.

Afin de réaliser des prévisions, il est nécessaire de vérifier que le modèle est bien en représentation canonique. Pour cela, on calcule le déterminant du polynôme $\Phi(B)$ et l'on regarde si les racines sont bien à l'extérieur du disque unité. Si tel est le cas, alors la prévision en $T+h$, notée $\hat{X}_T(h)$, du processus est :

$$\hat{X}_T(h) = \mathbb{E}[X_{T+h} | I_T] = \hat{c} + \hat{\Phi}_1 X_{T+h-1} + \dots + \hat{\Phi}_p X_{T+h-p} \quad (2.94)$$

où I_T désigne le passé de X jusqu'à la date T incluse.

Supposons que le processus VAR est stable, c'est - à - dire qu'il peut être représenté sous la forme d'un $VMA(\infty)$, et on a :

$$X_t = \sum_{j=0}^{\infty} \Psi_j \varepsilon_{t-j}, \quad \text{avec} \quad \Psi_0 = I_d \quad (2.95)$$

La prévision théorique est alors :

$$X_{T+h} = \sum_{j=0}^{\infty} \Psi_j \varepsilon_{T+h-j} \quad \text{avec} \quad \Psi_0 = I_d \quad (2.96)$$

$$X_T(h) = \mathbb{E}(X_{T+h}|I_T) = \Phi_h \varepsilon_t + \Psi_{h+1} \varepsilon_{t-1} + \Psi_{h+2} \varepsilon_{t-2} + \dots = \sum_{i \geq 0} \Psi_{h+i} \varepsilon_{t-i} \quad (2.97)$$

L'erreur de prévision en T à l'horizon h , noté $e_T(h)$, s'écrit sous la forme

$$e_T(h) = X_{T+h} - X_T(h) = \sum_{j=0}^{h-1} \Psi_j \varepsilon_{T+h-j} = [\Psi_{h-1} : \dots : \Psi_1 : I_d] \varepsilon_{T,h} \quad (2.98)$$

où $\varepsilon_{T,h} = (\varepsilon'_{T+1}, \dots, \varepsilon'_{T+h})'$. Puisque $\varepsilon_{T,h} \sim \mathcal{N}(0, I_h \otimes \Sigma_\varepsilon)$, par conséquent :

$$e_T(h) \sim \mathcal{N}(0, \Sigma_X(h)) \quad (2.99)$$

où $\Sigma_X(h) = \sum_{i=1}^{h-1} \Psi_i \Sigma_\varepsilon \Psi_i'$ est la matrice optimal de prévision.

En pratique, l'erreur de prévision estimée est donnée par :

$$\hat{e}_T(h) = X_{T+h} - \hat{X}_T(h) = \sum_{i=0}^{h-1} \hat{\Psi}_i \hat{\varepsilon}_{T+h-i} \quad (2.100)$$

$$\text{et } \hat{\Sigma}_X(h) = \sum_{i=1}^{h-1} \hat{\Psi}_i \hat{\Sigma}_\varepsilon \hat{\Psi}_i'$$

Exemple 2.6 Prédiction de investissement/revenu/consommation

Considérons notre exemple avec le jeu de données investment/income/consumption, on effectue une prévision de nos séries pour les horizons $h \geq 0$.

La prévision à $h = 1$ s'écrit :

$$\hat{X}_T(1) = \hat{c} + \hat{\Phi}_1 X_T + \hat{\Phi}_2 X_{T-1} \quad (2.101)$$

avec $X_{T-1} = X_{72}$ et $X_T = X_{73}$:

```
# Dernières valeurs pour l'échantillon d'apprentissage
df = pd.concat([DTrain.iloc[-2], DTrain.iloc[-1]], axis=1)
```

Table 2.4 – Valeurs de X_{T-1} et X_T

	1978-07-01	1978-10-01
dln_inv	0.026	0.036
dln_inc	0.024	0.005
dln_consump	0.013	0.006

Ainsi, on a pour $h = 1$:

$$\widehat{X}_T(1) = \hat{c} + \widehat{\Phi}_1 X_T + \widehat{\Phi}_2 X_{T-1} = \begin{pmatrix} -0.011 \\ 0.020 \\ 0.022 \end{pmatrix} \quad (2.102)$$

et pour $h = 2$,

$$\widehat{X}_T(2) = \hat{c} + \widehat{\Phi}_1 \widehat{X}_T(1) + \widehat{\Phi}_2 X_T = \begin{pmatrix} 0.011 \\ 0.020 \\ 0.015 \end{pmatrix} \quad (2.103)$$

et ainsi de suite.

Sous Python, La méthode `res.forecast()` effectue la prévision à l'horizon h .

```
# Pr vision du VAR(2)
DTest = dta.loc["1979-01-01":"1980-07-01",["dln_inv", "dln_inc", "dln_consump"]]
forecast = pd.DataFrame(
    results.forecast(DTrain.values[-results.k_ar:], steps=len(DTest)),
    columns=endog.columns, index = DTest.index)
forecast.insert(0, "h", range(1, len(DTest)+1))
```

Table 2.5 – Pr vision de investment/income/consumption   partir du mod le $VAR(2)$

	h	dln_inv	dln_inc	dln_consump
1979-01-01	1	-0.011	0.020	0.022
1979-04-01	2	0.011	0.020	0.015
1979-07-01	3	0.021	0.017	0.020
1979-10-01	4	0.012	0.021	0.019
1980-01-01	5	0.017	0.020	0.019
1980-04-01	6	0.017	0.020	0.020
1980-07-01	7	0.017	0.020	0.019

L'estimation de la matrice de variance covariance,   l'horizon h , se calcule comme suit :

$$\Sigma_{\widehat{X}}(h) = \Sigma_X(h) + \frac{1}{T}\Omega(h) \quad (2.104)$$

avec $\Sigma_X(h) = \sum_{i=0}^{h-1} \Psi_i \Sigma_\varepsilon \Psi_i'$. et $\Omega(h)$ est d fini par :

$$\Omega(h) = \mathbb{E} \left[\frac{\partial X_t(h)}{\partial \beta'} (\Gamma^{-1} \otimes \Sigma_\varepsilon) \frac{\partial X_t(h)'}{\partial \beta} \right] = \sum_{i=1}^{h-1} \sum_{j=1}^{h-1} \text{trace} \left[(B')^{h-1-i} \Gamma^{-1} B^{h-1-j} \Gamma \right] \Psi_i \Sigma_\varepsilon \Psi_j' \quad (2.105)$$

On constate que pour $h = 1$, on a $\Omega(1) = (dp + 1)\Sigma_\varepsilon$. Ainsi, l'approximation de $\Sigma_{\widehat{X}}(1)$ vaut :

$$\Sigma_{\widehat{X}}(1) = \Sigma_\varepsilon + \frac{dp + 1}{T}\Sigma_\varepsilon = \frac{T + dp + 1}{T}\Sigma_\varepsilon \quad (2.106)$$

$\Sigma_X(h)$ est estim e par :

$$\widehat{\Sigma}_X(h) = \sum_{i=0}^{h-1} \widehat{\Psi}_i \widehat{\Sigma}_\varepsilon \widehat{\Psi}_i' \quad (2.107)$$

où $\widehat{\Sigma}_\varepsilon$ est l'estimateur de Σ_ε . Ainsi, l'estimateur de $\Sigma_{\widehat{X}}(h)$ est :

$$\widehat{\Sigma}_{\widehat{X}}(h) = \widehat{\Sigma}_X(h) + \frac{1}{T}\widehat{\Omega}(h) \quad (2.108)$$

L'estimateur de la matrice de variance covariance se faire alors, au rang 1 , de la façon suivante :

$$\widehat{\Sigma}_{\widehat{X}}(1) = \frac{T + dp + 1}{T}\widehat{\Sigma}_\varepsilon = \frac{73 + 6 + 1}{73}\widehat{\Sigma}_\varepsilon = 10^{-4} \times \begin{pmatrix} 23.34 & 0.785 & 1.351 \\ 0.785 & 1.505 & 0.674 \\ 1.351 & 0.674 & 0.978 \end{pmatrix}$$

Les matrices de variance covariance à des ordres élevés se calculent à l'aide de la forme $VMA(\infty)$. La méthode `res.ma_rep` nous renvoie les coefficients Ψ_i .

```
# Coefficients VMA(2)
val = results.ma_rep(2)
print(val)

## [[ 1.          0.          0.          ]
##   [ 0.          1.          0.          ]
##   [ 0.          0.          1.          ]]
##
## [-0.31963183  0.14598513  0.96122876]
## [ 0.04393093 -0.15273112  0.28849916]
## [-0.00242299  0.22481339 -0.26396947]]
##
## [-0.05430212  0.26173998  0.41554235]
## [ 0.02857983  0.11376188 -0.08819013]
## [ 0.04517091  0.2608799  0.10998287]]]
```

Ainsi, pour $h = 2$, on a :

$$\widehat{\Sigma}_{\widehat{X}}(2) = \widehat{\Sigma}_X(2) + \frac{1}{T}\widehat{\Omega}(2) \quad \text{avec} \quad \widehat{\Sigma}_X(2) = \widehat{\Sigma}_\varepsilon + \widehat{\Psi}_1\widehat{\Sigma}_\varepsilon\widehat{\Psi}_1 \quad (2.109)$$

et

$$\widehat{\Omega}(2) = \text{tr} \left[(\widehat{B}'(ZZ')^{-1}\widehat{B}ZZ') \right] \widehat{\Sigma}_\varepsilon + \text{tr}(\widehat{B}')\widehat{\Sigma}_\varepsilon\widehat{\Psi}_1 + \text{tr}(\widehat{B})\widehat{\Psi}_1\widehat{\Sigma}_\varepsilon + \text{tr}(I_{dp+1})\widehat{\Psi}_1\widehat{\Sigma}_\varepsilon\widehat{\Psi}_1' \quad (2.110)$$

On obtient :

$$\widehat{\Sigma}_X(2) = \begin{pmatrix} 23.67 & 0.547 & 1.226 \\ 0.547 & 1.488 & 0.554 \\ 1.226 & 0.554 & 0.952 \end{pmatrix} \times 10^{-4}$$

et

$$\widehat{\Omega}(2) = \begin{pmatrix} 10.59 & 0.238 & 0.538 \\ 0.238 & 0.675 & 0.233 \\ 0.538 & 0.233 & 0.422 \end{pmatrix} \times 10^{-3}$$

Par conséquent :

$$\hat{\Sigma}_{\hat{X}}(2) = \begin{pmatrix} 25.12 & 0.580 & 1.300 \\ 0.580 & 1.581 & 0.586 \\ 1.300 & 0.586 & 1.009 \end{pmatrix} \times 10^{-4}$$

La méthode `res.forecast_cov()` nous renvoie les matrices de variance covariance $\hat{\Sigma}_X(h)$.

```
# Matrice de variance covariance
print(results.forecast_cov(steps= 2, method = "mse"))

## [[2.12962723e-03 7.16167908e-05 1.23239849e-04]
##    [7.16167908e-05 1.37337416e-04 6.14587508e-05]
##    [1.23239849e-04 6.14587508e-05 8.92036623e-05]]
##
## [[2.36738379e-03 5.47493288e-05 1.22614667e-04]
##    [5.47493288e-05 1.48822422e-04 5.54055772e-05]
##    [1.22614667e-04 5.54055772e-05 9.51582686e-05]]]
```

En supposant que les données soient générées par un processus gaussien, nous avons les intervalles de confiance suivante :

- $\widehat{invest}_T(1) \pm 1.96\hat{\sigma}_{invest}(1)$ ou -0.011 ± 0.095
- $\widehat{income}_T(1) \pm 1.96\hat{\sigma}_{income}(1)$ ou 0.020 ± 0.024
- $\widehat{cons}_T(1) \pm 1.96\hat{\sigma}_{cons}(1)$ ou 0.022 ± 0.019
- $\widehat{invest}_T(2) \pm 1.96\hat{\sigma}_{invest}(2)$ ou 0.011 ± 0.098
- $\widehat{income}_T(2) \pm 1.96\hat{\sigma}_{income}(2)$ ou 0.020 ± 0.025
- $\widehat{invest}_T(1) \pm 1.96\hat{\sigma}_{cons}(2)$ ou 0.015 ± 0.020

La méthode `res.forecast_interval()` nous renvoie les intervalles de confiance.

```
# Intervalle de confiance
var_pred=results.forecast_interval(y=DTrain.values[-results.k_ar:],
                                  steps=len(DTest),alpha=0.05)
mean_pred=pd.DataFrame(var_pred[0],index = DTest.index,
                       columns = ["dln_inv.mean","dln_inc.mean","dln_consump.mean"])
lower_pred=pd.DataFrame(var_pred[1],index = DTest.index,
                       columns = ["dln_inv.ci.lo","dln_inc.ci.lo","dln_consump.ci.lo"])
upper_pred=pd.DataFrame(var_pred[2],index = DTest.index,
                       columns = ["dln_inv.ci.up","dln_inc.ci.up","dln_consump.ci.up"])
var2_pred=pd.concat([mean_pred,lower_pred,upper_pred],axis=1)
var2_pred.insert(0,"h",range(1,len(DTest)+1))

# Représentation graphique de la prévision
fig,axes = plt.subplots(3,1,figsize=(16,10))
for idx, name in enumerate(DTrain.columns):
    DTrain[name].plot(ax=axes[idx],color = "black",label= f"{name} Train");
    DTest[name].plot(ax=axes[idx],color="gray",label = f"{name} test");
    mean_pred[f"{name}.mean"].plot(ax=axes[idx],color="red",
```

Table 2.6 – Pr vision du VAR(2)

	h	Pr�vision sur invest			Pr�vision sur income			Pr�vision sur consump		
		$\widehat{X}_{1,T}(h)$	[0.025	0.975]	$\widehat{X}_{2,T}(h)$	[0.025	0.975]	$\widehat{X}_{3,T}(h)$	[0.025	0.975]
1979-01-01	1	-0.011	-0.101	0.080	0.020	-0.003	0.043	0.022	0.003	0.040
1979-04-01	2	0.011	-0.085	0.106	0.020	-0.004	0.044	0.015	-0.004	0.034
1979-07-01	3	0.021	-0.075	0.117	0.017	-0.007	0.041	0.020	-0.001	0.041
1979-10-01	4	0.012	-0.085	0.109	0.021	-0.004	0.045	0.019	-0.003	0.040
1980-01-01	5	0.017	-0.080	0.114	0.020	-0.005	0.044	0.019	-0.002	0.040
1980-04-01	6	0.017	-0.080	0.114	0.020	-0.005	0.044	0.020	-0.002	0.041
1980-07-01	7	0.017	-0.080	0.114	0.020	-0.004	0.045	0.019	-0.002	0.041

```

label=f"{name} forecast");
lower_pred[f"{name}.ci.lo"].plot(ax=axes[idx],color="green",
                                label="lower bound",linestyle="--");
upper_pred[f"{name}.ci.up"].plot(ax=axes[idx],color="blue",
                                label="upper bound",linestyle="--");
axes[idx].fill_between(DTest.index,lower_pred[f"{name}.ci.lo"],
                      upper_pred[f"{name}.ci.up"],color='lightblue',alpha=0.5);
axes[idx].set_ylabel(f"{name}");
axes[idx].legend(bbox_to_anchor=(0.75, 1.15), ncol = 5);
axes[idx].set_xlabel("ann e");
plt.tight_layout();
plt.show()

```

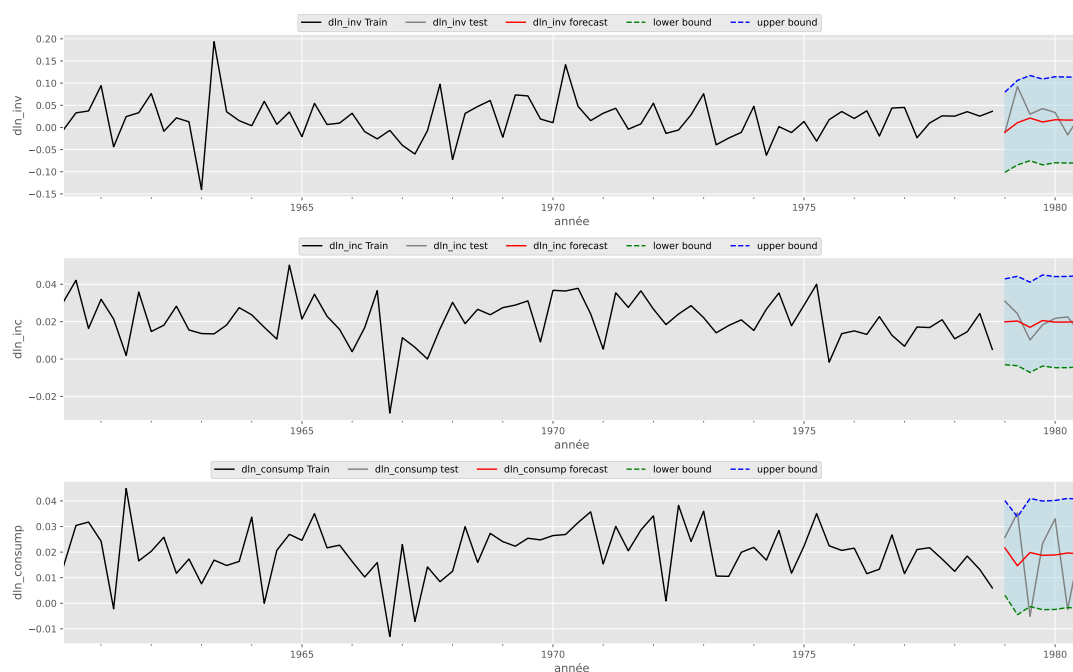


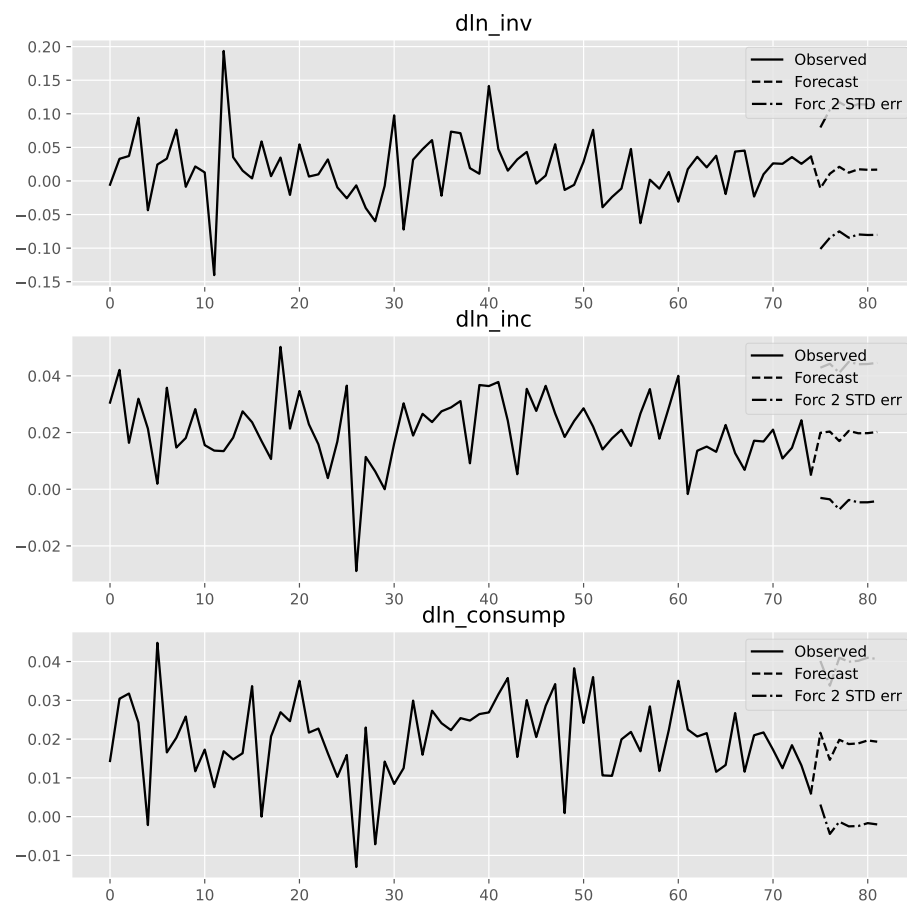
Figure 2.7 – Pr vision de investment/income/consumption

La m thode `res.plot_forecast()` permet de visualiser la pr vision d'un mod le VAR(p).

```

# Plot forecast
results.plot_forecast(len(DTest));
plt.show()

```

**Figure 2.8** – Prévision

Analyse structurelle d'un processus VAR

Sommaire

3.1 Concept de causalité	60
3.2 Analyse de réponse impulsionnelle	73

Lorsque nous tentons d'ajuster un processus d -dimensionnel $VAR(p)$, on désire plus qu'évaluer les paramètres de ce modèle. L'analyse VAR permet de déterminer les corrélations entre d variables. On peut aussi se permettre de déduire des relations dynamiques entre les variables du système, entre autres vérifier s'il existe des relations de cause à effet entre celles-ci. Nous ferons donc référence à l'analyse structurelle qui décrit un ensemble de ces méthodes. Dans la suite, nous allons présenter les concepts de causalité au sens de Granger et de causalité instantanée. Nous allons aussi présenter comment un changement survenu sur une variable peut en influencer une autre ainsi que la décomposition des covariances des erreurs de prédiction.

3.1 Concept de causalité

Soient $X_{1,t}$ et $X_{2,t}$ deux processus aléatoires et I_t un univers, c'est - à - dire une famille de processus contenant notamment $X_{1,t}$ et $X_{2,t}$. Désignons par \underline{I}_t l'information relative au passé :

$$\underline{I}_t = (I_s | s < t) \quad (3.1)$$

et soit $\underline{\underline{I}}_t$ l'information relative au passé et au présent :

$$\underline{\underline{I}}_t = (I_s | s \leq t) \quad (3.2)$$

De la même façon, on introduit les notations

$$\begin{cases} \underline{X}_{1,t} = (X_{1,s} | s < t) & \text{et} & \underline{\underline{X}}_{1,t} = (X_{1,s} | s \leq t) \\ \underline{X}_{2,t} = (X_{2,s} | s < t) & \text{et} & \underline{\underline{X}}_{2,t} = (X_{2,s} | s \leq t) \end{cases} \quad (3.3)$$

Si l'on suppose que tous les processus X_t et Y_t sont stationnaires, alors on a les définitions générales suivantes :

1. X_1 cause X_2 si l'erreur de prévision de X_2 est telle que

$$\sigma^2(X_{2,t}|\underline{I}_t) < \sigma^2(X_{2,t}|\underline{I}_t - \underline{X}_t) \quad (3.4)$$

où $\underline{I}_t - \underline{X}_t$ est l'information obtenue en retirant de \underline{I} les valeurs passées de X_2 ?

2. X_1 cause instantanément X_2 si

$$\sigma^2(X_{2,t}|\underline{I}_t, \underline{X}_{1,t}) < \sigma^2(X_{2,t}|\underline{I}_t) \quad (3.5)$$

où σ^2 désigne la variance de l'erreur de prévision.

La condition (1) signifie que, pour prévoir Y_t , la passé de X apporte une information supplémentaire par rapport à la seule prise en compte des autres variables figurant dans \underline{I} . La condition (2) signifie que la valeur présente de X apporte une information supplémentaire par rapport à la connaissance du passé des variables figurant dans \underline{I} .

Les inégalités sont toujours satisfaites. Elles se transforment en égalités si et seulement si :

1. X_1 ne cause pas X_2 si

$$\mathbb{E}(X_{2,t}|\underline{I}_t) = \mathbb{E}(X_{2,t}|\underline{I}_t - \underline{X}_{1,t}) \quad (3.6)$$

2. X_1 ne cause pas instantanément X_2 si

$$\mathbb{E}(X_{2,t}|\underline{I}_t, \underline{X}_{1,t}) = \mathbb{E}(X_{2,t}|\underline{I}_t) \quad (3.7)$$

La définition de la causalité est ici relative à un univers. On distingue trois grands types de causalité, dont le plus connu est la causalité au sens de Granger (1969).

3.1.1 Causalité au sens de Pierre et Haugh

Si $X_t = (X_{1,t}, X_{2,t})$ alors la causalité de X_1 vers X_2 peut être caractérisée par les corrélations des innovations des deux processus $X_{1,t}$ et $X_{2,t}$. Soit ε_t le processus des innovations de $X_{2,t}$ et η_t le processus des innovations de $X_{1,t}$. La fonction d'autocorrélation des deux processus d'innovations est donnée par :

$$\rho_{\eta\varepsilon}(h) = \frac{\text{Cov}(\eta_t, \varepsilon_{t-h})}{\sigma_\eta \sigma_\varepsilon} \quad (3.8)$$

Dans ce cas, X_2 ne cause pas X_1 si $\rho_{\eta\varepsilon}(h) = 0 \forall h > 0$ (ou $\forall h \geq 0$ si la causalité instantanée est exclue). L'innovation de X_1 doit être non corrélée avec toutes les innovations passées associées au processus $X_{2,t}$.

On peut donner une autre caractérisation de cette non-causalité en termes de régression par le biais de la notion de causalité au sens de Granger.

3.1.2 Causalité au sens de Granger

De façon heuristique, on dira que X_1 cause X_2 si la prévision de X_2 fondée sur la connaissance des passés conjoints de X_1 et de X_2 est meilleure que la prévision fondée sur la seule connaissance du passé de X_2 . Plus formellement, on retiendra la définition suivante de la causalité au sens de Granger :

1. X_1 cause X_2 à la date t si :

$$\mathbb{E}(X_{2,t}|\underline{X}_{2,t-1}, \underline{X}_{1,t-1}) \neq \mathbb{E}(X_{2,t}|\underline{X}_{2,t-1}) \quad (3.9)$$

2. X_1 cause instantanément X_2 à la date t si :

$$\mathbb{E}(X_{2,t}|\underline{X}_{2,t-1}, \underline{X}_{1,t}) \neq \mathbb{E}(X_{2,t}|\underline{X}_{2,t-1}, \underline{X}_{1,t-1}) \quad (3.10)$$

3. X_1 ne cause pas X_2 à la date t si :

$$V\varepsilon(X_{2,t}|\underline{X}_{2,t-1}, \underline{X}_{1,t-1}) = V\varepsilon(X_{2,t}|\underline{X}_{1,t-1}) \quad (3.11)$$

où $V\varepsilon$ désigne la matrice de variance covariance de l'erreur de prévision, $\underline{X}_{1,t} = \{X_{1,t-i}, i \geq 0\}$, $\underline{X}_{1,t-1} = \{X_{1,t-i}, i \geq 1\}$ et $\underline{X}_{2,t-1} = \{X_{2,t-i}, i \geq 1\}$.

Rappelons que, dans le cas linéaire, l'opérateur d'espérance conditionnelle désigne la meilleure prévision linéaire d'une variable fondée sur un certain ensemble d'informations. Si l'on suppose que les prévisions sont effectuées par régression linéaire, l'opérateur d'espérance conditionnelle représente une fonction de régression. Ainsi $\mathbb{E}(X_{2,t}|\underline{X}_{2,t-1}, \underline{X}_{1,t-1})$ désigne la régression linéaire de $X_{2,t}$ sur son passé ($\underline{X}_{2,t-1}$) et sur le passé de $X_{1,t}$ jusqu'à la date $t-1$ incluse ($\underline{X}_{1,t-1}$). De même, la notation $V\varepsilon(X_{2,t}|\underline{X}_{2,t-1}, \underline{X}_{1,t-1})$ désigne la matrice de variance covariance de l'erreur de prévision associée à la régression linéaire de Y_t sur son passé ($\underline{X}_{2,t-1}$) et sur le passé de $X_{1,t}$ jusqu'à la date $t-1$ incluse ($\underline{X}_{1,t-1}$).

3.1.2.1 Causalité au sens de Granger dans un système bivarié

Le cadre le plus simple pour l'étude de la causalité au sens de Granger est celui d'un système constitué uniquement de deux variables X_1 et X_2 . Il permet normalement de se familiariser aisément avec les outils et représentations qui seront mobilisés dans des cadres plus généraux.

Nous verrons tout d'abord ce qui implique l'absence de causalité selon Granger d'une variable X_1 vers une variable X_2 sur les coefficients du VAR afférent au couple $(X_{1,t}, X_{2,t})'$. Dans un second temps, nous montrerons l'équivalence des approches VAR ou VMA pour traiter de la causalité selon Granger dans ce cadre bivarié.

Soit $X_t = (X_{1,t}, X_{2,t})'$. X_t est supposé stable, donc stationnaire. On a donc une écriture de Wold :

$$X_t = \mu_X + \Psi(B)\varepsilon_t \quad (3.12)$$

où ε_t est un processus bruit blanc. De plus, de part son inversibilité, on a également une écriture VAR , éventuellement d'ordre infini :

$$\Phi(B)X_t = c + \varepsilon_t \quad (3.13)$$

avec $\Phi(B) = I - \Phi_1 B - \Phi_2 B^2 + \dots = \Psi(B)^{-1}$ et $c = \Psi(B)^{-1}\mu_X$.

Si X_1 n'est pas un prédictor avancé de X_2 , d'après la définition, les coefficients des X_1 retardés dans l'équation de X_2 doivent être nuls. En effet, si tel est le cas, la présence ou l'absence de l'historique de X_1 dans cette équation ne modifie évidemment pas la prévision que l'on peut faire de X_2 , qui est alors tirée d'un processus univarié de type AR . L'absence de causalité de Granger revient donc à imposer des contraintes de nullité sur les coefficients hors-diagonaux des matrices Φ_i , $i = 1, 2, \dots$. Ainsi, pour un $VAR(1)$, on aura :

$$\begin{pmatrix} X_{1,t} \\ X_{2,t} \end{pmatrix} = \begin{pmatrix} c_0^1 \\ c_0^2 \end{pmatrix} + \begin{pmatrix} \phi_{1,1}^1 & \phi_{1,2}^1 \\ \phi_{2,1}^1 & \phi_{2,2}^1 \end{pmatrix} \begin{pmatrix} X_{1,t-1} \\ X_{2,t-1} \end{pmatrix} + \begin{pmatrix} \varepsilon_{1,t} \\ \varepsilon_{2,t} \end{pmatrix} \quad (3.14)$$

et :

$$\begin{aligned}
X_1 \nrightarrow X_2 &\Leftrightarrow \phi_{2,1}^1 = 0 \\
&\Leftrightarrow \Phi(B) = \begin{pmatrix} 1 - \phi_{1,1}^1 B & -\phi_{1,2}^1 B \\ 0 & 1 - \phi_{2,2}^1 B \end{pmatrix} \\
X_2 \nrightarrow X_1 &\Leftrightarrow \phi_{1,2}^1 = 0 \\
&\Leftrightarrow \Phi(B) = \begin{pmatrix} 1 - \phi_{1,1}^1 B & 0 \\ -\phi_{2,1}^1 B & 1 - \phi_{2,2}^1 B \end{pmatrix}
\end{aligned}$$

On en déduit donc :

- $X_1 \rightarrow X_2 \Leftrightarrow \phi_{2,1}^1 \neq 0$
- $X_2 \rightarrow X_1 \Leftrightarrow \phi_{1,2}^1 \neq 0$

Considérons à présent un VAR(2) bivarié :

$$\begin{pmatrix} X_{1,t} \\ X_{2,t} \end{pmatrix} = \begin{pmatrix} c_0^1 \\ c_0^2 \end{pmatrix} + \begin{pmatrix} \phi_{1,1}^1 & \phi_{1,2}^1 \\ \phi_{2,1}^1 & \phi_{2,2}^1 \end{pmatrix} \begin{pmatrix} X_{1,t-1} \\ X_{2,t-1} \end{pmatrix} + \begin{pmatrix} \phi_{1,1}^2 & \phi_{1,2}^2 \\ \phi_{2,1}^2 & \phi_{2,2}^2 \end{pmatrix} \begin{pmatrix} X_{1,t-2} \\ X_{2,t-2} \end{pmatrix} + \begin{pmatrix} \varepsilon_{1,t} \\ \varepsilon_{2,t} \end{pmatrix} \quad (3.15)$$

avec,

$$\begin{aligned}
X_1 \nrightarrow X_2 &\Leftrightarrow \phi_{2,1}^1 = \phi_{2,1}^2 = 0 \\
&\Leftrightarrow \Phi(B) = \begin{pmatrix} 1 - \phi_{1,1}^1 B - \phi_{1,1}^2 B^2 & -\phi_{1,2}^1 B - \phi_{1,2}^2 B^2 \\ 0 & 1 - \phi_{2,2}^1 B - \phi_{2,2}^2 B^2 \end{pmatrix} \\
X_2 \nrightarrow X_1 &\Leftrightarrow \phi_{1,2}^1 = \phi_{1,2}^2 = 0 \\
&\Leftrightarrow \Phi(B) = \begin{pmatrix} 1 - \phi_{1,1}^1 B - \phi_{1,1}^2 B^2 & 0 \\ -\phi_{2,1}^1 B - \phi_{2,1}^2 B^2 & 1 - \phi_{2,2}^1 B - \phi_{2,2}^2 B^2 \end{pmatrix}
\end{aligned}$$

Ce qui implique que :

- $X_1 \rightarrow X_2 \Leftrightarrow \phi_{2,1}^1 \neq 0$ ou $\phi_{2,1}^2 \neq 0$
- $X_2 \rightarrow X_1 \Leftrightarrow \phi_{1,2}^1 \neq 0$ ou $\phi_{1,2}^2 \neq 0$

Plus généralement, pour un VAR d'ordre p quelconque, si on partitionne $\Phi(B)$ conformément au découpage pris sur X_t , il vient :

$$X_1 \nrightarrow X_2 \Leftrightarrow \Phi(B) = \begin{pmatrix} 1 - \sum_{j=1}^p \phi_{1,1}^j B^j & -\sum_{j=1}^p \phi_{1,2}^j B^j \\ 0 & 1 - \sum_{j=1}^p \phi_{2,2}^j B^j \end{pmatrix} \quad (3.16)$$

$$X_2 \nrightarrow X_1 \Leftrightarrow \Phi(B) = \begin{pmatrix} 1 - \sum_{j=1}^p \phi_{1,1}^j B^j & 0 \\ -\sum_{j=1}^p \phi_{2,1}^j B^j & 1 - \sum_{j=1}^p \phi_{2,2}^j B^j \end{pmatrix} \quad (3.17)$$

Dans l'un ou l'autre cas, $\Phi(B)$ est une matrice triangulaire, triangulaire supérieure lorsque $X_1 \nrightarrow X_2$, triangulaire inférieure lorsque $X_2 \nrightarrow X_1$.

3.1.2.2 Conséquence de la non causalité pour la représentation VMA

Si le VAR est stable, on peut basculer vers sa représentation VMA infinie : $\Phi(B)X_t = c + \varepsilon_t \Rightarrow X_t = \mu_X + \Psi(B)\varepsilon_t$, avec $\Psi(B) = \Phi(B)^{-1}$. Or l'inverse d'une triangulaire est une triangulaire de même nature. Dans ces conditions :

$$X_t = \begin{pmatrix} X_{1,t} \\ X_{2,t} \end{pmatrix} = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix} + \Psi(B) \begin{pmatrix} \varepsilon_{1,t} \\ \varepsilon_{2,t} \end{pmatrix} \quad (3.18)$$

et

$$X_1 \nrightarrow X_2 \Leftrightarrow \Psi(B) = \begin{pmatrix} 1 + \sum_{i=1}^{\infty} \psi_{1,1}^i B^i & \sum_{i=1}^{\infty} \psi_{1,2}^i B^i \\ 0 & 1 + \sum_{i=1}^{\infty} \psi_{2,2}^i B^i \end{pmatrix} \quad (3.19)$$

$$X_2 \nrightarrow X_1 \Leftrightarrow \Psi(B) = \begin{pmatrix} 1 + \sum_{i=1}^{\infty} \psi_{1,1}^i B^i & 0 \\ \sum_{i=1}^{\infty} \psi_{2,1}^i B^i & 1 + \sum_{i=1}^{\infty} \psi_{2,2}^i B^i \end{pmatrix} \quad (3.20)$$

Ainsi, dans un VAR bivarié, on a l'ensemble des équivalences suivantes dans le cas où $X_1 \nrightarrow X_2$:

1. $\phi_{2,1}^i = 0, i = 0, \dots, p$: les coefficients des X_1 retardés dans l'équation de X_2 sont nuls
2. $\psi_{2,1}^i = 0, i = 1, \dots, \infty$: La fonction de réponse de X_2 au choc $\varepsilon_{1,t}$ est nulle
3. La totalité de la variance des erreurs de prévision commises sur X_2 pour n'importe quel horizon est attribuable à $\varepsilon_{2,t}$.

Naturellement, le même type de conclusion est obtenu dans le cas symétrique où $X_2 \nrightarrow X_1$.

3.1.3 Causalité au sens de Sims

Sims (1980) a introduit un concept de causalité légèrement différent. Il propose de considérer les valeurs futures de $X_{2,t}$. Si les valeurs futures de X_2 peuvent permettre d'expliquer les valeurs présentes de X_1 , alors X_1 est la cause de X_2 . De façon similaire, on dira que X_1 cause X_2 si les innovations de X_1 contribuent à la variance de l'erreur de prévision de X_2 . Considérons un processus $VAR(p)$ bivarié :

$$\begin{cases} X_{1,t} = c_0^1 + \sum_{i=1}^p \phi_{1,1}^i X_{1,t-i} + \sum_{i=1}^p \phi_{1,2}^i X_{2,t-i} + \sum_{i=1}^p \gamma_1^i X_{2,t+i} + \varepsilon_{1,t} \\ X_{2,t} = c_0^2 + \sum_{i=1}^p \phi_{2,1}^i X_{1,t-i} + \sum_{i=1}^p \phi_{2,2}^i X_{2,t-i} + \sum_{i=1}^p \gamma_2^i X_{1,t+i} + \varepsilon_{2,t} \end{cases} \quad (3.21)$$

Dans ce cas :

- X_1 ne cause pas X_2 si l'hypothèse nulle $\gamma_2^1 = \gamma_2^2 = \dots = \gamma_2^p = 0$ est acceptée.
- X_2 ne cause pas X_1 si l'hypothèse nulle $\gamma_1^1 = \gamma_1^2 = \dots = \gamma_1^p = 0$ est acceptée.

Il s'agit d'un test de Fisher de nullité des coefficients.

3.1.4 Mesure et test de causalité

3.1.4.1 Mesures de causalité

A partir de la définition de la causalité au sens de Granger, il est possible de définir des mesures de causalité :

1. Mesure de causalité de X_1 vers X_2 :

$$C_{X_1 \rightarrow X_2} = \log \left(\frac{\det V\varepsilon(X_{2,t} | \underline{X}_{2,t-1})}{\det V\varepsilon(X_{2,t} | \underline{X}_{2,t-1}, \underline{X}_{1,t-1})} \right) \quad (3.22)$$

Bien évidemment, si X_1 ne cause pas X_2 alors, $C_{X_1 \rightarrow X_2} = 0$. Dans le cas contraire, on a $C_{X_1 \rightarrow X_2} > 0$.

2. Mesure de causalité instantanée de X_1 vers X_2 :

$$C_{X_1 \leftrightarrow X_2} = \log \left(\frac{\det V\varepsilon(X_{2,t} | \underline{X}_{2,t-1})}{\det V\varepsilon(X_{2,t} | \underline{X}_{2,t-1}, \underline{X}_{1,t})} \right) \quad (3.23)$$

3.1.4.2 Tests de non causalité

Il est possible de tester l'hypothèse nulle de non causalité au moyen de la statistique du rapport du maximum de vraisemblance :

$$LR = TC_{X_1 \rightarrow X_2} \quad (3.24)$$

Sous l'hypothèse nulle, cette statistique suit une loi de χ^2 à $N(T - N)p$ degrés de liberté où N est le nombre de contraintes imposées. La règle de décision est la suivante :

- Si $LR < \chi^2_{(N(T-N)p)}$, on accepte l'hypothèse nulle d'absence de causalité.
- Si $LR \geq \chi^2_{(N(T-N)p)}$, on rejette l'hypothèse nulle en faveur de l'hypothèse alternative de présence de causalité.

Exemple 3.1

Considérons le processus VAR(p) à deux variables $X_{1,t}$ et $X_{2,t}$ suivant :

$$\begin{pmatrix} X_{1,t} \\ X_{2,t} \end{pmatrix} = \begin{pmatrix} c_0^1 \\ c_0^2 \end{pmatrix} + \begin{pmatrix} \phi_{1,1}^1 & \phi_{1,2}^1 \\ \phi_{2,1}^1 & \phi_{2,2}^1 \end{pmatrix} \begin{pmatrix} X_{1,t-1} \\ X_{2,t-1} \end{pmatrix} + \dots + \begin{pmatrix} \phi_{1,1}^p & \phi_{1,2}^p \\ \phi_{2,1}^p & \phi_{2,2}^p \end{pmatrix} \begin{pmatrix} X_{1,t-p} \\ X_{2,t-p} \end{pmatrix} + \begin{pmatrix} \varepsilon_{1t} \\ \varepsilon_{2t} \end{pmatrix} \quad (3.25)$$

Tester l'absence de causalité de $X_{1,t}$ vers $X_{2,t}$ revient à effectuer un test de restriction sur les coefficients des variables X_t de la représentation VAR. En effet :

- $X_{1,t}$ ne cause pas $X_{2,t}$ si l'hypothèse nulle suivante est acceptée : $H_0 : \phi_{2,1}^j = 0, j = 1, \dots, p$.
- $X_{2,t}$ ne cause pas $X_{1,t}$ si l'hypothèse nulle suivante est acceptée : $H_0 : \phi_{1,2}^j = 0, j = 1, \dots, p$.

Les tests sont alors des tests de Fisher classiques ou de khi - deux. On les effectue soit équation par équation, soit directement par comparaison entre un modèle VAR contraint et un modèle VAR non contraint.

Remarque 3.1

Si l'on est amené à rejeter les deux hypothèses nulles, on a une causalité bi - directionnelle, on parle de boucle rétroactive (feedback effect).

Proposition 3.1 *Test de Wald pour la non causalité au sens de Granger*

Dans le contexte d'un modèle VAR(p) estimé, on peut conduire des tests de causalité en remplaçant les paramètres testés par leurs estimateurs des moindres carrés ou du maximum de vraisemblance. Pour identifier l'absence de causalité au sens de Granger, il faut vérifier la nullité de certaines valeurs (ou blocs de valeurs) des matrices Φ_1, \dots, Φ_p . Ici nous utilisons le test de Wald. En effet, on veut tester :

$$\begin{cases} H_0 : R\beta = c \\ H_1 : R\beta \neq c \end{cases} \quad (3.26)$$

où R est une matrice de taille $(N \times (d^2p + d))$ et de rang N et c un vecteur de taille $(N \times 1)$. N est le nombre de contraintes à tester.

Soit β , le vecteur de coefficients à estimer et $\hat{\beta}$, un estimateur de β et supposons que

$$\sqrt{T}(\hat{\beta} - \beta) \xrightarrow[T \rightarrow \infty]{\mathcal{L}} \mathcal{N}(0, \Gamma^{-1} \otimes \Sigma_\varepsilon) \quad (3.27)$$

Si les matrices ZZ'/T et $\hat{\Sigma}_\varepsilon$ convergent en probabilité respectivement vers Γ et Σ_ε et sont non singulières et si $H_0 : R\beta = c$ est vraie, où R est une matrice de dimension $(N \times (d^2p + d))$ et de rang N , alors la statistique de Wald λ_W définie par

$$\lambda_W = T(R\hat{\beta} - c)' [R(\Gamma^{-1} \otimes \Sigma_\varepsilon)R']^{-1} (R\hat{\beta} - c) \quad (3.28)$$

suit sous H_0 une loi de χ^2 à N degrés de liberté.

Le fait que cette statistique soit distribuée selon une loi de χ^2 découle du fait que l'équation (3.27) implique

$$\sqrt{T}(R\hat{\beta} - R\beta) \xrightarrow[T \rightarrow \infty]{\mathcal{L}} \mathcal{N}(0, R(\Gamma^{-1} \otimes \Sigma_\varepsilon)R') \quad (3.29)$$

Proposition 3.2 *Test de Fisher pour la non causalité au sens de Granger*

Si on tient compte du fait que les matrices Γ^{-1} et Σ_ε sont estimées, on peut ajuster la statistique de Wald et obtenir une nouvelle statistique λ_F définie par :

$$\lambda_F = \lambda_W / N \quad (3.30)$$

Cette statistique suit sous H_0 une loi de Fisher à N et $d(T - dp - 1)$ degrés de liberté.

On comparera donc ces statistiques à leurs valeurs critiques afin de rejeter ou non l'hypothèse nulle de non - causalité au sens de Granger qui est testée.

Exemple 3.2 *Test de causalité au sens de Granger*

Sous Python, la méthode `res.test_causality()` permet de tester la causalité au sens de Granger. Cette méthode possède un paramètre `kind` qui permet de spécifier le type de statistique à utiliser (fisher ou wald).

— **Test de causalité de Granger basé sur la statistique de Fisher λ_F**

```
# Extraction des résultats du test de causalité au sens de Granger
def granger_fisher_output(res):
    title = res.title + res.signif_str
    print("====*23)
    print(title)
    print("====*23)
    print(res.h0)
    print("====*23)
    df = pd.DataFrame({
        "Test statistic": res.test_statistic, "Critical value" : res.crit_value,
        "ddl1" : res.df[0], "ddl2" : res.df[1], "p-value" : res.pvalue,
        "signif" : res.signif}, index = [str(res.test)])
    print(df)
    print("====*23)
    print(res.conclusion_str)
    print("====*23)
```

— H_0 : (dl_inc, dln_consump) ne cause pas au sens de Granger dln_inv

```
## Granger Causality : Fisher test
# H_0: ['dln_inc', 'dln_consump'] do not Granger-cause dln_inv
g1_f = results.test_causality(caused='dln_inv', kind='f',
                             causing = ["dln_inc", "dln_consump"])
granger_fisher_output(g1_f)

## =====
## Granger causality F-test at 5% significance level
## =====
## H_0: ['dln_inc', 'dln_consump'] do not Granger-cause dln_inv
## =====
##          Test statistic  Critical value  ddl1  ddl2  p-value  signif
## granger          1.591716          2.417258    4   198  0.177963    0.05
## =====
## Conclusion: fail to reject H_0
## =====
```

La statistique λ_F vaut 1.592 avec une pvalue supérieure au seuil critique de 5%, par conséquent, on ne rejette pas l'hypothèse nulle.

— H_0 : (dln_inv, dln_consump) ne cause pas au sens de Granger dln_inc

```
## Granger Causality : Fisher test
# H_0: ['dln_inv', 'dln_consump'] do not Granger-cause dln_inc
g2_f = results.test_causality(caused='dln_inc', kind='f',
                             causing = ["dln_inv", "dln_consump"])
granger_fisher_output(g2_f)
```



```
## =====
## Granger causality F-test at 5% significance level
## =====
## H_0: ['dln_inv', 'dln_consump'] do not Granger-cause dln_inc
## =====
##          Test statistic Critical value  ddl1  ddl2  p-value  signif
## granger          1.946573          2.417258    4   198  0.104217    0.05
## =====
## Conclusion: fail to reject H_0
## =====
```

Au seuil de 5%, on ne peut rejeter l'hypothèse nulle. La statistique λ_F vaut 1.947 et la pvalue associée est supérieure au seuil critique de 5%.

— H_0 : (dl_inv,dl_inc) ne cause pas au sens de Granger dln_consump

```
## Granger Causality : Fisher test
# H_0: ['dln_inv', 'dln_inc'] do not Granger-cause dln_consump
g3_f = results.test_causality(caused='dln_consump',kind='f',
                              causing = ["dln_inv", "dln_inc"])
granger_fisher_output(g3_f)
```

```
## =====
## Granger causality F-test at 5% significance level
## =====
## H_0: ['dln_inv', 'dln_inc'] do not Granger-cause dln_consump
## =====
##          Test statistic Critical value  ddl1  ddl2  p-value  signif
## granger          3.774607          2.417258    4   198  0.00555    0.05
## =====
## Conclusion: reject H_0
## =====
```

La pvalue est inférieure au seuil critique de 5%, donc on rejette l'hypothèse nulle.

— **Test de causalité de Granger basé sur la statistique de Wald λ_W**

On refait le test en se basant sur la statistique λ_W qui suit une χ^2 .

```
# Extraction du test de causalité de Granger
def granger_wald_output(res):
    title = res.title + res.signif_str
    print("="*61)
    print(title)
    print("="*61)
    print(res.h0)
    print("="*61)
    df = pd.DataFrame({
        "Test statistic": res.test_statistic,"Critical value" : res.crit_value,
        "df" : res.df,"p-value" : res.pvalue,
        "signif" : res.signif},index = [str(res.test)])
    print(df)
```

```
print("="*61)
print(res.conclusion_str)
print("="*61)
```

— H_0 : (dln_inc, dln_consump) ne cause pas au sens de Granger dln_inv

```
## Granger Causality : Wald test
# H_0: ['dln_inc', 'dln_consump'] do not Granger-cause dln_inv
g1_w = results.test_causality(caused='dln_inv',kind='wald',
                              causing = ["dln_inc", "dln_consump"])
granger_wald_output(g1_w)
```

```
## =====
## Granger causality Wald-test at 5% significance level
## =====
## H_0: ['dln_inc', 'dln_consump'] do not Granger-cause dln_inv
## =====
##          Test statistic Critical value  df   p-value  signif
## granger          6.366863          9.487729   4  0.173375    0.05
## =====
## Conclusion: fail to reject H_0
## =====
```

La statistique λ_W vaut 6.367 avec une pvalue supérieure au seuil critique de 5%, par conséquent, on ne rejette pas l'hypothèse nulle.

— H_0 : (dln_inv, dln_consump) ne cause pas au sens de Granger dln_inc

```
## Granger Causality : Wald test
# H_0: ['dln_inv', 'dln_consump'] do not Granger-cause dln_inc
g2_w = results.test_causality(caused='dln_inc',kind='wald',
                              causing = ["dln_inv", "dln_consump"])
granger_wald_output(g2_w)
```

```
## =====
## Granger causality Wald-test at 5% significance level
## =====
## H_0: ['dln_inv', 'dln_consump'] do not Granger-cause dln_inc
## =====
##          Test statistic Critical value  df   p-value  signif
## granger          7.78629          9.487729   4  0.099728    0.05
## =====
## Conclusion: fail to reject H_0
## =====
```

Au seuil de 5%, on ne peut rejeter l'hypothèse nulle. La statistique λ_W vaut 7.786 et la pvalue associée est supérieure au seuil critique de 5%.

— H_0 : (dln_inv, dln_inc) ne cause pas au sens de Granger dln_consump

```
## Granger Causality : Wald test
# H_0: ['dln_inv', 'dln_inc'] do not Granger-cause dln_consump
g3_w = results.test_causality(caused='dln_consump', kind='wald',
                             causing = ["dln_inv", "dln_inc"])
granger_wald_output(g3_w)

## =====
## Granger causality Wald-test at 5% significance level
## =====
## H_0: ['dln_inv', 'dln_inc'] do not Granger-cause dln_consump
## =====
##          Test statistic  Critical value  df    p-value  signif
## granger          15.098428          9.487729   4    0.004501    0.05
## =====
## Conclusion: reject H_0
## =====
```

La pvalue est inférieure au seuil critique de 5%, donc on rejette l'hypothèse nulle.

Une meilleure façon de mesurer la causalité au sens de Granger entre les différentes variables est de la tester de façon individuelle pour chaque variable en fonction d'une autre.

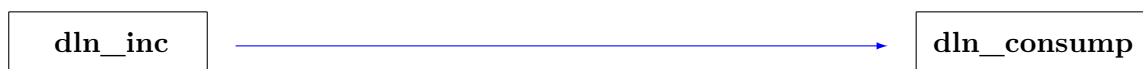
```
# Test de causalité au sens de Granger
def grangers_causation_matrix(model, **kwargs):
    stat_pvalue = dict()
    for r in model.names:
        for c in model.names:
            res = model.test_causality(caused=r, causing=c, **kwargs)
            stat_pvalue[(r,c)] = (res.test_statistic, res.pvalue)
    df = pd.DataFrame(stat_pvalue, index = ["statistic", "p-value"]).T.round(4)
    df = (df.reset_index()
          .rename(columns={"level_0": "caused", "level_1": "causing"}))
    return df
# Application de la fonction
granger_var = grangers_causation_matrix(results)
```

Table 3.1 – Test de causalité au sens de Granger

caused	causing	statistic	p-value
dln_inv	dln_inv	3.4786	0.0327
dln_inv	dln_inc	0.0485	0.9527
dln_inv	dln_consump	1.5004	0.2256
dln_inc	dln_inv	1.7683	0.1733
dln_inc	dln_inc	0.6850	0.5053
dln_inc	dln_consump	1.7184	0.1820
dln_consump	dln_inv	0.9715	0.3803
dln_consump	dln_inc	6.1465	0.0026
dln_consump	dln_consump	2.0441	0.1322

Le tableau (3.1) montre qu'on a une seule corrélation au sens de Granger qui soit significative : $\Delta \ln(inc) \rightarrow \Delta \ln(consump)$. Ce résultat était prévisible au regard de la significativité des coefficients associés à dln_inc dans l'équation de $dln_consump$.

Nous pouvons résumer le test de causalité de Granger entre les variables `dln_inv`, `dln_inc` et `dln_consump` en faisant un schéma pour montrer clairement le sens de causalité entre elles.



Causalité au sens de Granger entre `dln_inv`/`dln_inc`/`dln_consump`

3.1.4.3 La non causalité instantanée

La non causalité instantanée est un concept différent de la non causalité de Granger. Elle regarde si pour un instant du temps, c'est - à - dire en t , deux ou plusieurs variables évoluent de manière indépendante c'est - à - dire si un choc sur une période n'a pas de répercussion instantanée sur les autres variables. On devine sans qu'il soit nécessaire de faire de développements mathématiques que ce sera le cas si les innovations du processus sont indépendantes. Plus précisément, il n'y aura pas de causalité instantanée entre $X_{1,t}$ et $X_{2,t}$ si ε_t et $\varepsilon_{2,t}$ sont non corrélés, c'est - à - dire $\mathbb{E}(\varepsilon_{1,t}\varepsilon_{2,t}) = 0$. Un test similaire de Wald est proposé pour tester la causalité instantanée.

Exemple 3.3 Causalité instantanée entre *cons*, *income* et *invest*

Sous Python, la méthode `res.test_inst_causality()` permet de tester la causalité instantanée entre les variables.

```
# Extraction des résultats du test causalité instantanée
def instantaneous_output(res):
    title = res.title + res.signif_str
    print("=="*29)
    print(title)
    print("=="*29)
    print(res.h0)
    print("=="*29)
    df = pd.DataFrame({
        "Test statistic": res.test_statistic, "Critical value" : res.crit_value,
        "df" : res.df, "p-value" : res.pvalue, "signif" : res.signif},
        index = [str(res.test)])
    print(df)
    print("=="*29)
    print(res.conclusion_str)
    print("=="*29)
```

— H_0 : `dln_inv` ne cause pas instantanément (`dln_inc`, `dln_consump`)

```
# H_0 : dln_inv do not instantaneous cause (dln_inc, dln_consump)
inst1 = results.test_inst_causality(causing = "dln_inv", signif=0.05)
instantaneous_output(inst1)

## =====
## Instantaneous causality Wald-test at 5% significance level
## =====
## H_0: dln_inv does not instantaneously cause [dln_inc, dln_consump]
## =====
##      Test statistic  Critical value  df    p-value  signif
```

```
## inst          5.458868          5.991465    2  0.065256    0.05
## =====
## Conclusion: fail to reject H_0
## =====
```

La p-value est supérieure au seuil critique de 5%, donc on accepte l'hypothèse nulle, par conséquent `dln_inv` ne cause pas instantanément (`dln_inc`, `dln_consump`).

— H_0 : `dln_inc` ne cause pas instantanément (`dln_inv`, `dln_consump`)

```
# H_0 : dln_inc do not instantaneous cause (dln_inv, dln_consump)
inst2 = results.test_inst_causality(causing = "dln_inc", signif=0.05)
instantaneous_output(inst2)
```

```
## =====
## Instantaneous causality Wald-test at 5% significance level
## =====
## H_0: dln_inc does not instantaneously cause [dln_inv, dln_consump]
## =====
##      Test statistic  Critical value  df   p-value  signif
## inst          17.231064          5.991465    2  0.000181    0.05
## =====
## Conclusion: reject H_0
## =====
```

La p - value est inférieure au seuil critique de 5%, donc on rejette l'hypothèse nulle : `dln_inc` cause instantanément (`dln_inv`, `dln_consump`).

— H_0 : `dln_consump` ne cause pas instantanément (`dln_inv`, `dln_inc`)

```
# H_0 : dln_consump do not instantaneous cause (dln_inv, dln_inc)
inst3 = results.test_inst_causality(causing = "dln_consump", signif=0.05)
instantaneous_output(inst3)
```

```
## =====
## Instantaneous causality Wald-test at 5% significance level
## =====
## H_0: dln_consump does not instantaneously cause [dln_inv, dln_inc]
## =====
##      Test statistic  Critical value  df   p-value  signif
## inst          19.040702          5.991465    2  0.000073    0.05
## =====
## Conclusion: reject H_0
## =====
```

La p - value est inférieure au seuil critique de 5%, donc on rejette l'hypothèse nulle : `dln_consump` cause instantanément (`dln_inv`, `dln_inc`).

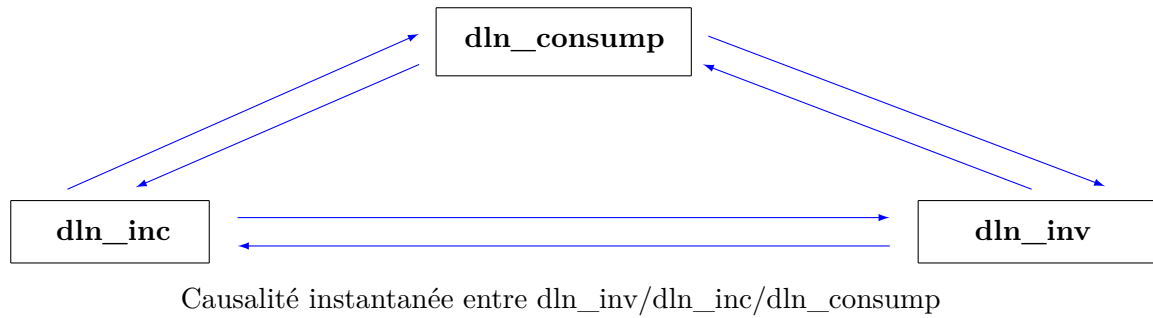
Pour avoir la causalité instantanée d'une variable à l'autre, nous affichons la matrice de variance covariance des résidus estimés. L'attribut `res.sigma_u` :

```
# Matrice des corrélations des résidus estimés
sigma_u = pd.DataFrame(results.sigma_u, index = results.names,
                        columns = results.names)
```

Table 3.2 – Matrice de variance - covariance des résidus estimés

	dln_inv	dln_inc	dln_consump
dln_inv	0.0021296	0.0000716	0.0001232
dln_inc	0.0000716	0.0001373	0.0000615
dln_consump	0.0001232	0.0000615	0.0000892

Etant donnée que $\hat{\Sigma}_\varepsilon$ ne comporte pas d'élément nul, on en déduit que la causalité instantanée est significative pour l'ensemble des variables. On a le schéma suivant :



3.1.5 Exogénéité

Considérons le processus VAR(p) défini précédemment et écrit sous la forme matricielle :

$$\begin{pmatrix} X_{1,t} \\ X_{2,t} \end{pmatrix} = \begin{pmatrix} c_0^1 \\ c_0^2 \end{pmatrix} + \begin{pmatrix} \phi_{1,1}^1 & \phi_{1,2}^1 \\ \phi_{2,1}^1 & \phi_{2,2}^1 \end{pmatrix} \begin{pmatrix} X_{1,t-1} \\ X_{2,t-1} \end{pmatrix} + \dots + \begin{pmatrix} \phi_{1,1}^p & \phi_{1,2}^p \\ \phi_{2,1}^p & \phi_{2,2}^p \end{pmatrix} \begin{pmatrix} X_{1,t-p} \\ X_{2,t-p} \end{pmatrix} + \begin{pmatrix} \varepsilon_{1,t} \\ \varepsilon_{2,t} \end{pmatrix} \quad (3.31)$$

Le bloc de variables $(X_{1,t-1}, X_{1,t-2}, \dots, X_{1,t-p})$ est exogène par rapport au bloc de variables $(X_{2,t-1}, X_{2,t-2}, \dots, X_{2,t-p})$ si le fait de rajouter le bloc $X_{1,t}$ n'améliore pas significativement la détermination des variables $X_{2,t}$. On effectue un test de restrictions sur les coefficients des variables $X_{1,t}$ de la représentation VAR. Plus précisément, en accord avec Engle, Hendry, and Richard (1983), une variable est dite strictement exogène si ses valeurs à chaque période sont indépendantes des valeurs des perturbations aléatoires de toutes les périodes¹. Le concept de stricte exogénéité est ainsi lié au concept de causalité au sens de Granger (1969).

3.2 Analyse de réponse impulsionnelle

Dans les applications empiriques, une des principales utilisations des processus VAR réside dans l'analyse de réponse impulsionnelle. Il s'agit ici de préciser l'impact d'un choc exogène sur les valeurs et futures des variables constituant le VAR. Il s'agit par exemple d'apprécier l'impact sur les valeurs futures de l'ensemble des composantes de X d'une augmentation de 1 point dans le niveau de l'une de ces variables. Étant donné la structure d'un VAR, ces chocs d'innovation ne peuvent provenir que d'une modification imposée dans une des composantes du vecteur ε .

1. Notons que le concept de prédétermination est lié à la notion d'exogénéité. Une variable est dite prédéterminée en t si toutes ses valeurs présentes et passées sont indépendantes des erreurs présentes et si celles-ci ne sont pas autocorrélées.

3.2.1 Fonctions de réponse simples

Ces réponses présentes et futures du système X à des chocs de type ε sont aisément identifiées à partir de l'écriture $VMA(\infty)$:

$$X_t = \mu + \sum_{i=0}^{+\infty} \Psi_i \varepsilon_{t-i} \quad \text{avec} \quad \Psi_0 = I \quad (3.32)$$

Et donc :

$$\frac{\partial X_t}{\partial \varepsilon_{t-h}} = \Psi_h \quad (3.33)$$

Ainsi, un choc d'amplitude unitaire en t se produisant sur la $j^{\text{ième}}$ composante de ε_t impacte la $i^{\text{ième}}$ composante de X_{t+h} d'un montant égal à $\psi_{i,j}^h$ qui est la dérivée partielle de $X_{i,t+h}$ par rapport à $\varepsilon_{j,t}$. Habituellement on représente ces fonctions de réponses simples en portant en abscisse les valeurs de h et en ordonnée les valeurs estimées des coefficients $\psi_{i,j}^h$ ².

Remarque 3.2

L'idée générale de l'analyse des fonctions de réponse aux innovations est de résumer l'information concernant l'évolution d'une composante $X_{k,t}$ suite à une impulsion sur $X_{i,t}$, à la date $t = 0$, en supposant que toutes les autres variables sont constantes pour $t \leq 0$.

Exemple 3.4 Fonction de réponse impulsionnelle simple

Les coefficients de la forme $VMA(\infty)$ sont les coefficients des réponses impulsionnelles. On a donc :

```
# Coefficients des réponses impulsionnelles
def impulresponse(res,lag):
    print("Estimates of impulse responses")
    ma_rep = res.ma_rep(lag)
    for i in range(len(ma_rep)):
        print(f"stage i = {i}")
        print(ma_rep[i])
# Application
impulresponse(results,3)
```

```
## Estimates of impulse responses
## stage i = 0
## [[1. 0. 0.]
##  [0. 1. 0.]
##  [0. 0. 1.]]
## stage i = 1
## [[-0.31963183  0.14598513  0.96122876]
##  [ 0.04393093 -0.15273112  0.28849916]
```

2. Nous noterons que la masse d'informations à exploiter devient vite problématique : dans un VAR de dimension d , il existe donc d innovations dont on peut étudier les réponses et d variables qui répondent, soit d^2 fonctions de réponse.

```
## [-0.00242299  0.22481339 -0.26396947]]
## stage i = 2
## [[-0.05430212  0.26173998  0.41554235]
## [ 0.02857983  0.11376188 -0.08819013]
## [ 0.04517091  0.2608799  0.10998287]]
## stage i = 3
## [[ 0.11903613  0.35283688 -0.40789324]
## [-0.00884347  0.07147085  0.11976609]
## [-0.00055082 -0.0981804  0.09096104]]
```

On visualise ces fonctions de réponses impulsionnelles.

```
# Fonction de réponse impulsionnelle simple
irf = results.irf(8)
irf.plot(orth=False);
plt.show()
```

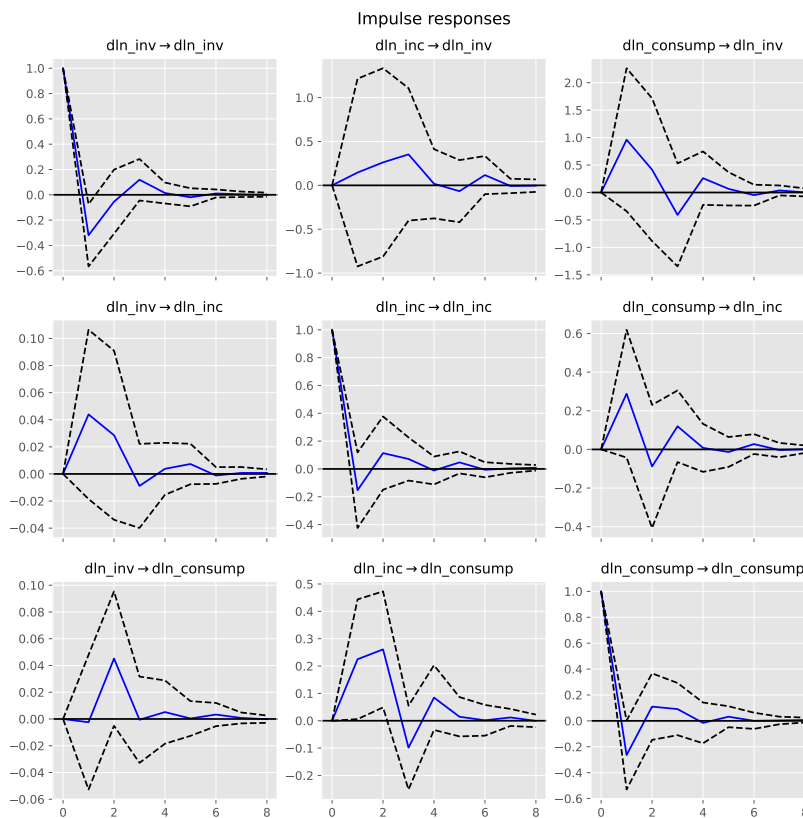


Figure 3.1 — Analyse des fonctions de réponse simple

La figure 3.1 présente les fonctions de réponse impulsionnelles (notamment celles des réponses non orthogonalisées). Sur le graphique $\Delta \ln(inc) \rightarrow \Delta \ln(consump)$, on voit que $dln_consump$ s'accroît en réponse d'un choc de dln_inc .

On peut vouloir se concentrer uniquement sur une seule variable pour analyser les chocs, dans ce cas, on exécute la commande suivante :


```
# IRF : invest
irf.plot(impulse='dln_inv');
plt.show()
```

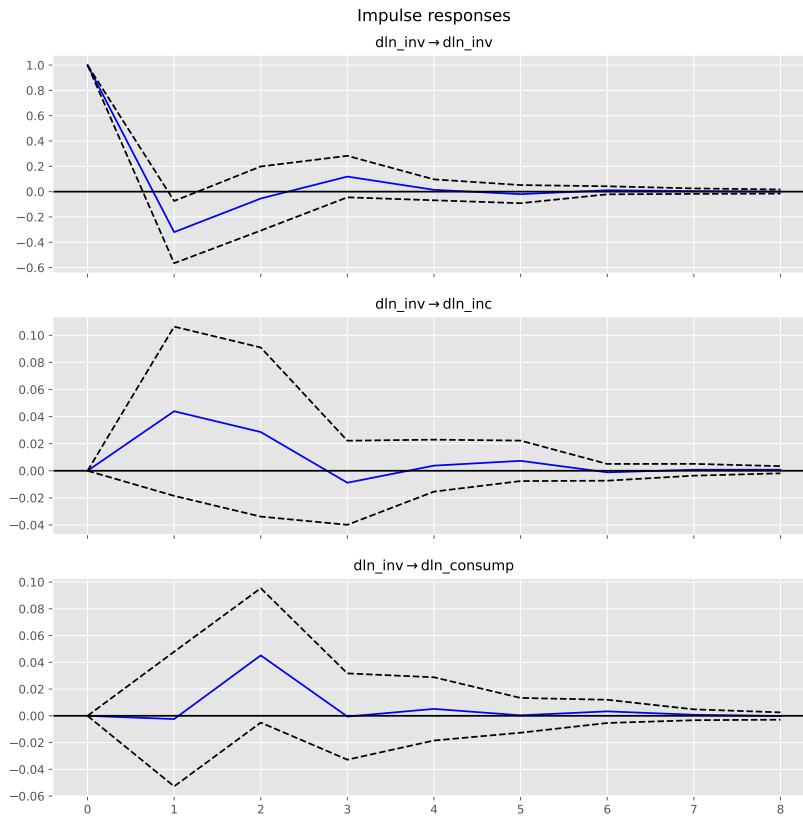


Figure 3.2 – Estimated responses of all to a forecast error impulse in invest with estimated asymptotic two-standard error bounds

La figure 3.2 illustre les réponses de l'investissement aux impulsions. Sur le graphe $\Delta \ln(consump) \rightarrow \Delta \ln(inv)$, aucune réponse n'est significative. Ce résultat est conforme à l'analyse de causalité de Granger de $dln_inc/dln_consump$ à dln_inv .

On peut également considérer les réponses accumulées. Ces dernières donnent également une certaine information sur l'analyse des chocs. Pour cela, on considère les matrices définie par $\Lambda_h = I_d + \Psi_1 + \Psi_2 + \dots + \Psi_h$, avec la matrice limite $\Lambda_\infty = I_d + \Psi_1 + \Psi_2 + \dots + \Psi_h + \dots = (I_d - \Phi_1 - \dots - \Phi_p)^{-1}$. Les fonctions de réponses accumulées renforcent la recherche des réponses impulsionnelles individuelles.

```
# Coefficients cumulés de réponse
for i in range(1,5):
    lambd = results.ma_rep(i).sum(axis=0)
    print(f"Matrices des réponses cumulées au retard i = {i}")
    print(lambd)

## Matrices des réponses cumulées au retard i = 1
## [[ 0.68036817  0.14598513  0.96122876]
##   [ 0.04393093  0.84726888  0.28849916]
##   [-0.00242299  0.22481339  0.73603053]]
```

```

## Matrices des réponses cumulées au retard i = 2
## [[0.62606605 0.40772511 1.3767711 ]
## [0.07251076 0.96103077 0.20030903]
## [0.04274792 0.48569329 0.84601341]]
## Matrices des réponses cumulées au retard i = 3
## [[0.74510218 0.76056199 0.96887786]
## [0.06366729 1.03250162 0.32007512]
## [0.0421971 0.38751289 0.93697445]]
## Matrices des réponses cumulées au retard i = 4
## [[0.75943519 0.77862479 1.23011763]
## [0.06745862 1.02137527 0.32808398]
## [0.04736554 0.4720873 0.92121128]]

```

```

# Effet cumulatif
irf.plot_cum_effects(orth=False);
plt.show()

```

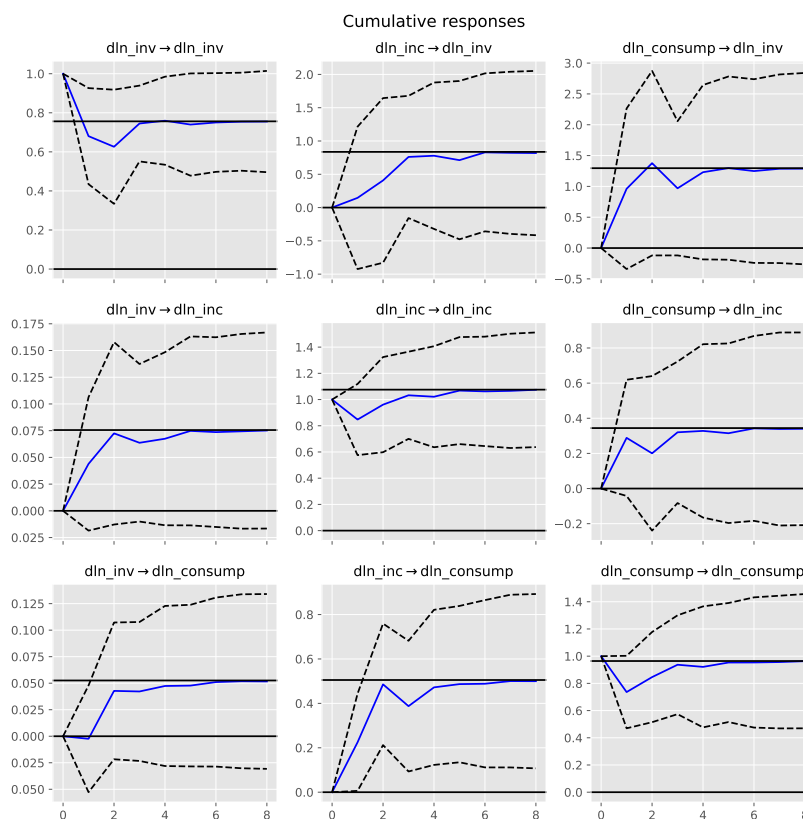


Figure 3.3 – Accumulated and long-run responses with estimated asymptotic two-standard error bounds

Il existe cependant une difficulté : la mesure de l'impact d'une innovation au moyen de la dérivée partielle qui permet de retrouver l'interprétation usuelle que l'on donne à un multiplicateur³ suppose que les autres innovations soient maintenues à zéro. En effet, si en mettant une innovation non nulle dans la $j^{\text{ième}}$ composante de ε_t , cela génère également une déviation dans la $i^{\text{ième}}$ composante, alors la réponse de X va cumuler les deux impacts et ne pourra donc pas être attribuée en totalité à la seule $j^{\text{ième}}$ composante. Or les composantes de ε_t n'ayant aucune raison

3. A savoir la mesure de l'impact de la variation d'une variable toutes autres variables inchangées.

d'être de covariance nulle, on ne peut pas écarter ce risque. Ce n'est que lorsque Σ_ε est diagonale que l'effet multiplicateur d'une innovation peut être mesuré par la dérivée partielle.

La solution passe par la substitution d'un ensemble d'aléatoires orthogonales aux ε_t initiaux. Il sera alors possible d'étudier la réponse du système à ces nouveaux chocs via les fonctions de réponse orthogonalisées.

3.2.2 Fonctions de réponse orthogonalisées

Il existe plusieurs possibilités de réaliser une telle orthogonalisation des chocs ε_t . Sims (1980) utilise une décomposition de Cholesky de la matrice Σ_ε . cette méthode de décomposition via une matrice triangulaire inférieure unique est depuis la plus populaire. Elle repose sur le fait que toute matrice définie - positive peut s'écrire comme un produit PP' où P est une matrice inversible à éléments nuls au-dessus de la diagonale et à éléments positifs sur la diagonale⁴. Comme Σ_ε est une matrice de variance covariance, elle est définie positive. On est donc assuré de l'existence d'une telle matrice P de dimension $(d \times d)$ unique telle que :

$$\Sigma_\varepsilon = PP' \quad (3.34)$$

Soit $\omega_t = P^{-1}\varepsilon_t$ un nouvel ensemble de d aléatoires construits à partir de combinaisons linéaires des ε_t initiaux, combinaisons dont les poids sont donnés par les éléments de P^{-1} . Il vient que :

$$\begin{cases} \mathbb{E}[\omega_t] = \mathbb{E}[P^{-1}\varepsilon_t] = P^{-1}\mathbb{E}[\varepsilon_t] = 0 \\ \Sigma_\omega = \mathbb{E}[\omega_t\omega_t'] = P^{-1}\mathbb{E}[\varepsilon_t\varepsilon_t'] (P^{-1})' = P^{-1}\Sigma_\varepsilon(P^{-1})' = P^{-1}PP'(P')^{-1} = I \end{cases} \quad (3.35)$$

En d'autres termes, les ω_t ainsi construites sont des aléatoires centrées, de variance unitaire et orthogonales entre elles⁵.

Pour obtenir les fonctions de réponse orthogonalisées afférentes aux chocs ω_t , il suffit de reprendre leur expression de définition $\omega_t = P^{-1}\varepsilon_t$ pour obtenir $\varepsilon_t = P\omega_t$ et de remplacer ε_t par cette expression dans la formulation du $VMA(\infty)$. Il vient :

$$X_t = \Phi^{-1}(B)[c + \varepsilon_t] = m + \Psi(B)\varepsilon_t = m + \Psi(B)P\omega_t = m + \Xi(B)\omega_t \quad (3.36)$$

avec $\Xi(B) = \Psi(B)P$.

Grâce à l'orthogonalisation, on peut modifier une des composantes de ω_t tout en maintenant les autres composantes à une valeur constante, notamment zéro : le raisonnement toutes autres variables inchangées est ici valide, et la dérivée partielle mesure bien l'effet multiplicateur du seul choc que l'on provoque. En pratique, les coefficients de ces fonctions de réponse sont naturellement donnés par

$$\frac{\partial X_{it}}{\partial \omega_{j,t-h}} = \hat{\xi}_{i,j}^h \quad (3.37)$$

Ici, $\hat{\xi}_{i,j}^h$ est l'estimation⁶ de l'impact d'un choc unitaire dans la $j^{\text{ième}}$ composante de ω_t sur la valeur de la $i^{\text{ième}}$ composante de X_{t+h} . Comme pour les fonctions simples, les fonctions de

4. Cette contrainte de positivité assure l'unicité de T .

5. On montre aussi aisément que $\mathbb{E}[\omega_t\omega_{t-s}'] = 0$ si $s \neq 0$: ω_t est, de même que ε_t , un processus en bruit blanc.

6. Ces valeurs estimées sont obtenues en plusieurs étapes : l'estimation du $VAR(p)$ fournit les matrices \hat{A}_j $j = 1, \dots, p$. Celles-ci sont injectées dans les équations de passage de l'écriture VAR vers le VMA donnant ainsi les $\hat{\Psi}_h$, $h = 1, 2, \dots$. La décomposition de $\hat{\Sigma}_\varepsilon$, également estimée à la première étape, aboutit à \hat{P} , et finalement, $\hat{\Xi}_h = \hat{\Psi}_h\hat{P}$, $h = 1, 2, \dots$.

réponse orthogonalisées sont souvent représentées graphiquement avec h en abscisse et $\hat{\xi}_{i,j}^h$ en ordonnée et permettent d'apprécier la propagation dans le temps des effets d'un choc sur chacune des variables du système étudié.

Enfin, rappelons-nous que dans un système stationnaire les effets des chocs sont transitoires : lorsque h augmente on doit tendre vers un multiplicateur nul. Ceci doit être observé aussi bien sur les fonctions de réponse simples que sur les réponses orthogonalisées.

Maintenant que les chocs dans ces fonctions de réponse orthogonalisées sont représentés par des valeurs attribuées aux ω , il importe de connaître leur significations.

Exemple 3.5 Chocs orthogonalisés de investment/income/consumption

En utilisant la décomposition de Cholesky sur $\hat{\Sigma}_\varepsilon$, on obtient :

```
# Décomposition de Cholesky
P = np.linalg.cholesky(results.sigma_u)
print(P)

## [[0.04614788 0.          0.          ]
##  [0.0015519  0.0116159  0.          ]
##  [0.00267054 0.00493413 0.00759778]]
```

Ainsi, les coefficients $\hat{\Xi}_j$ sont obtenus grâce à la formule $\hat{\Xi}_j = \hat{\Psi}_j P$.

```
# Coefficient xi1
xi1 = np.dot(results.ma_rep(2)[1],P)
print(xi1)

## [[-0.01195678  0.00643858  0.0073032 ]
##  [ 0.00256075 -0.00035062  0.00219195]
##  [-0.00046787  0.00130895 -0.00200558]]
```

Une façon rapide pour obtenir les coefficients $\hat{\Xi}_i$ est d'utiliser la méthode `res.orth_ma_rep()`. Pour les 3 périodes antérieures, on a :

```
# Coefficients xi
def impulresponseorth(res,lag):
    print("Estimates of responses to orthogonal innovations")
    val = results.orth_ma_rep(maxn=lag).round(4)
    for i in range(len(val)):
        print("stage i = {i+1}")
        print(val[i])
# Application
impulresponseorth(results,3)

## Estimates of responses to orthogonal innovations
## stage i = {i+1}
## [[0.0461 0.          0.          ]
##  [0.0016 0.0116 0.          ]
```

```

## [0.0027 0.0049 0.0076]]
## stage i = {i+1}
## [[-0.012  0.0064  0.0073]
## [ 0.0026 -0.0004  0.0022]
## [-0.0005  0.0013 -0.002 ]]
## stage i = {i+1}
## [[-0.001  0.0051  0.0032]
## [ 0.0013  0.0009 -0.0007]
## [ 0.0028  0.0036  0.0008]]
## stage i = {i+1}
## [[ 0.005  0.0021 -0.0031]
## [ 0.      0.0014  0.0009]
## [ 0.0001 -0.0007  0.0007]]

# Fonction de réponse impulsionnelle orthogonalisée
irf.plot(orth=True);
plt.show()

```

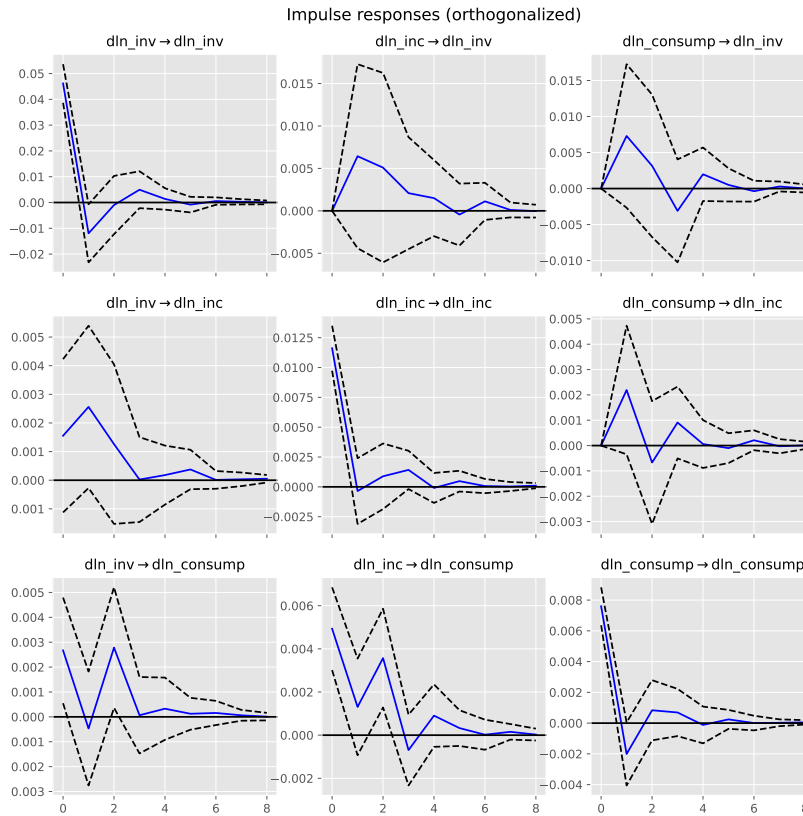


Figure 3.4 – Analyse des fonctions de réponse orthogonalisée

3.2.3 Interprétation des chocs orthogonalisés et conséquence de cette orthogonalisation

On considère la i^{ime} équation d'un $VAR(p)$ afférant à un X de dimension d :

$$X_{i,t} = c_0^i + \sum_{j=1}^{j=d} \phi_{i,j}^1 X_{j,t-1} + \sum_{j=1}^{j=d} \phi_{i,j}^2 X_{j,t-2} + \cdots + \sum_{j=1}^{j=d} \phi_{i,j}^p X_{j,t-p} + \varepsilon_{i,t} \quad (3.38)$$

soit encore dans sa version centrée

$$X_{i,t} - \mu_i = \sum_{j=1}^{j=d} \phi_{i,j}^1 (X_{j,t-1} - \mu_j) + \sum_{j=1}^{j=d} \phi_{i,j}^2 (X_{j,t-2} - \mu_j) + \cdots + \sum_{j=1}^{j=d} \phi_{i,j}^p (X_{j,t-p} - \mu_j) + \varepsilon_{i,t} \quad (3.39)$$

On peut concevoir que les fonctions de réponse précédentes apprécient les conséquences d'un choc en comparant la dynamique générée par une innovation à une situation de référence telle que, en l'absence d'innovation, le système se reproduirait à l'identique période après période. Clairement cette position d'équilibre est donnée par

$$X_{t-1} = X_{t-2} = \cdots = X_{t-p} = \mu$$

puisque, alors, en l'absence de chocs, c'est-à-dire pour $\varepsilon_t = 0$, nous aurions

$$X_t - \mu = X_{t+1} - \mu = X_{t+2} - \mu = \cdots = 0$$

Dès lors, si à une date t_0 nous fixons $\varepsilon_{i,t_0} = 1$ et $\varepsilon_{i,t_0+k} = 0$, $k = 1, 2, \dots$, alors on va impliquer sur X_i et aussi sur les autres variables des trajectoires qui vont dévier de zéro et qui donc représenteront l'impact du choc unitaire effectué en t_0 , mais aussi les impacts des modifications que cela aura entraîné dans d'autres composantes de ε dès lors que Σ_ε n'est plus diagonale : les covariances non nulles interdiront de maintenir à zéro ces autres composantes lorsque ε_i prendra la valeur 1. C'est précisément la raison qui pousse à l'orthogonalisation. Le seul intérêt du raisonnement précédent est de faire comprendre que ce qu'on appelle innovation à une date t_0 pour une équation quelconque d'un *VAR* consiste obligatoirement en une modification de son résidu en t_0 . En conséquence, si nous construisons un choc tel qu'il n'affecte pas ε_{i,t_0} , alors X_i ne peut pas répondre en t_0 . Naturellement les valeurs postérieures à t_0 de X_i pourront être affectées⁷, mais la réaction contemporaine sera nulle.

Précisément l'orthogonalisation de Cholesky crée des chocs ω qui génèrent de telles configurations. En effet, nous savons que $\varepsilon = P\omega$ avec P matrice triangulaire inférieure. On a donc :

$$\begin{pmatrix} \varepsilon_{1,t} \\ \vdots \\ \varepsilon_{d,t} \end{pmatrix} = \underbrace{\begin{pmatrix} p_{1,1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ p_{d,1} & \cdots & p_{d,d} \end{pmatrix}}_{=P} \begin{pmatrix} \omega_{1,t} \\ \vdots \\ \omega_{d,t} \end{pmatrix} \quad (3.40)$$

soit encore :

$$\begin{cases} \varepsilon_{1,t} = p_{1,1}\omega_{1,t} \\ \varepsilon_{2,t} = p_{2,1}\omega_{1,t} + p_{2,2}\omega_{2,t} \\ \vdots \\ \varepsilon_{d,t} = p_{d,1}\omega_{1,t} + p_{d,2}\omega_{2,t} + \cdots + p_{d,d}\omega_{d,t} \end{cases} \quad (3.41)$$

Ainsi, une innovation dans $\omega_{1,t}$ pourra affecter l'ensemble des ε en t et donc une réaction contemporaine dans l'ensemble des variables. En revanche, une innovation dans $\omega_{2,t}$ n'affecte pas $\varepsilon_{1,t}$

7. puisque une ou plusieurs autres variables présentes dans la liste des explicatives de X_i seront affectées en t_0 .

et donc ne provoquera pas de réaction contemporaine dans $X_{1,t}$. Par contre elle peut affecter les $\varepsilon_{i,t}$ tels que $i \geq 2$ et donc il sera possible d'observer une réponse dans $X_{2,t}, X_{3,t}, \dots, X_{d,t}$. Enfin, pour les mêmes raisons, une modification de $\omega_{d,t}$ ne peut pas affecter immédiatement $X_{1,t}, X_{2,t}, \dots, X_{(d-1),t}$, seules $X_{d,t}$ et les explicatives de rangs supérieurs dans l'ordre d'entrée pourront répondre à l'instant même du choc.

Plus généralement, l'aléatoire $\omega_{i,t}$ de rang i ne pourra pas provoquer de réponse contemporaine dans les variables $X_{j,t}$ de rang inférieur à i , i.e. telles que $j < i$ mais qu'elle est susceptible d'avoir un impact immédiat sur les $X_{j,t}$ de rang supérieur ou égal à i . Nous devons bien comprendre que cette absence de réponse immédiate de certaines variables n'est absolument une propriété réelle du système étudié : c'est une conséquence de la méthode de décomposition de Cholesky qui est seulement liée à l'ordre d'entrée des variables dans le VAR⁸. En d'autres termes, les valeurs des fonctions de réponse dépendent de l'ordre d'entrée des variables dans le vecteur X .

La question qui se pose alors naturellement est de savoir s'il existe un ordre d'entrée optimal, et, si oui, comment le déterminer ? Si on se limite aux précédents développements⁹, nous avons un élément de réponse qui malheureusement en pratique se révèle souvent difficile à appliquer. Il faut entrer en premier la variable dont les modifications sont les plus susceptibles d'affecter de façon contemporaine l'ensemble des autres, et en dernier celle qui est le moins à même d'avoir un impact immédiat sur toutes celles qui la précèdent. Par exemple, supposons que nous soyons intéressé par la modélisation des ventes d'une entreprise et que vous pensiez que celles-ci dépendent de la conjoncture nationale ou/et régionale, représentée chacune par un indicateur conjoncturel. Nous pourrions alors construire X comme :

$$X_t = \begin{pmatrix} \text{indicateur national}_t \\ \text{indicateur régional}_t \\ \text{ventes de l'entreprise}_t \end{pmatrix}$$

Cependant, dans beaucoup de cas il sera difficile de justifier l'ordre choisit. En pratique, la prudence recommande d'essayer quelques permutations sur l'ordre d'entrée initial et de comparer les fonctions de réponse ainsi obtenues afin de vérifier la robustesse des conclusions que nous pourrions être amené à énoncer sur la seule base du classement initial¹⁰.

Outre la difficulté précédemment soulevée, une autre interrogation se pose sur l'orthogonalisation : si les chocs dont on étudie les réponses sont les aléatoires ω , il reste à comprendre la signification : étudier l'impact de chocs dont on ne sait pas ce qu'ils signifient n'est pas particulièrement intéressant. Pour clarifier les choses, on peut repartir des équations de reconstruction des ε à partir des ω en nous limitant à un système de dimension 3, la généralisation à des systèmes supérieurs étant immédiate :

$$\begin{cases} \varepsilon_{1,t} = p_{1,1}\omega_{1,t} \\ \varepsilon_{2,t} = p_{2,1}\omega_{1,t} + p_{2,2}\omega_{2,t} \\ \varepsilon_{3,t} = p_{3,1}\omega_{1,t} + p_{3,2}\omega_{2,t} + p_{3,3}\omega_{3,t} \end{cases}$$

D'après la première équation, $\omega_{1,t}$ n'est rien d'autre que $\varepsilon_{1,t}$ réduite. Comme cette dernière est de variance σ_1^2 et que $\omega_{1,t}$ est de variance unitaire, on a immédiatement la valeur du coefficient

8. Évitions par exemple d'interpréter cette absence de réponse immédiate comme la preuve de l'absence de dépendance instantanée entre les variables concernées.

9. Les VAR structurels constituent un autre moyen de répondre à cette interrogation.

10. Souvenons-nous qu'en présence de d variables dans le vecteur X , $d!$ permutations sont possibles, ce qui peut rendre l'exercice rapidement impraticable, à moins de ne retenir que quelques-unes d'entre elles en se fondant sur des considérations théoriques telles que, par exemple, la plus ou moins grande capacité d'une variable à affecter instantanément les autres variables du système.

de normalisation $p_{1,1} = \sigma_1$: un choc unitaire sur $\omega_{1,t}$ correspond à un choc d'un écart-type sur $\varepsilon_{1,t}$. Pour la deuxième, on peut encore l'écrire comme

$$\varepsilon_{2,t} = \frac{p_{2,1}}{p_{1,1}} \varepsilon_{1,t} + p_{2,2} \omega_{2,t} \quad (3.42)$$

Comme $\omega_{2,t}$ est orthogonal à $\omega_{1,t}$ et donc à $\varepsilon_{1,t}$, $p_{2,2}\omega_{2,t}$ est donc le résidu de la projection de $\varepsilon_{2,t}$ sur $\varepsilon_{1,t}$: c'est la partie de $\varepsilon_{2,t}$ non expliquée linéairement par $\varepsilon_{1,t}$. Ainsi, $\omega_{1,t}$ et $\omega_{2,t}$ engendrent le même espace que $\varepsilon_{1,t}$ et $\varepsilon_{2,t}$. Dans ces conditions, comme $\omega_{3,t}$ est orthogonal à $\omega_{1,t}$ et $\omega_{2,t}$, $p_{3,3}\omega_{3,t}$ est le résidu de la projection de $\varepsilon_{3,t}$ sur cet espace, et c'est donc la partie de $\varepsilon_{3,t}$ non expliquée linéairement par $\varepsilon_{1,t}$ et $\varepsilon_{2,t}$.

En généralisant ce raisonnement, $\omega_{i,t}$ apparaît donc comme étant la partie de $\varepsilon_{i,t}$ non expliquée linéairement par les $\varepsilon_{j,t}$ de rangs inférieurs à i .

3.2.4 Intervalle de confiance des fonctions de réponses

L'analyse économique des fonctions de réponse n'est pertinente que si l'on dispose du degré de précision de l'estimation. Comme à chaque fois que l'on dispose d'une estimation, il est nécessaire d'avoir une idée de sa précision, via sa variance afin, par exemple, de pouvoir mener des tests d'hypothèses où construire des intervalles de confiance. Dans le cas qui nous intéresse ici, avant de commenter la forme des fonctions de réponse estimées, il est utile de statuer sur l'hypothèse de leur nullité : il serait dangereux de raconter une histoire sur l'impact d'un choc (signe de l'impact, propagation dans le temps), si dans le même temps on ne peut pas rejeter une hypothèse de nullité de ces coefficients de réponse. Il est donc nécessaire de construire des intervalles de confiance autour des coefficients estimés pour en particulier savoir si 0 est ou n'est pas dans l'intervalle en question. Pour construire ceux-ci on a recourt soit à la méthode delta, soit à des intervalles bootstrappés. Cependant, il est généralement admis que la première conduit à des intervalles trop larges et beaucoup de publications préfèrent utiliser des intervalles de confiance obtenus au moyen d'une méthode bootstrap.

3.2.4.1 Intervalles obtenus via la distribution asymptotique et la méthode delta

Avec les précédents calculs, on dispose des estimateurs ponctuels de la réponse de la variable X_i à un choc ε_j ou ω_j après h périodes via $\hat{\psi}_{i,j}^h$ ou $\hat{\xi}_{i,j}^h$. On rappelle que $\Psi(B) = \Phi(B)^{-1}$ et que les estimateurs des matrices autorégressives $\hat{\Phi}_i$ $i = 1, \dots, p$ sont des estimateurs du maximum de vraisemblance. Ainsi, d'après la propriété d'invariance de ceux-ci, les estimateurs du maximum de vraisemblance de $\Psi(B)$ sont donc $\hat{\Psi}(B) = \hat{\Phi}(B)^{-1}$. Ils sont convergents, asymptotiquement gaussiens.

Soit $\hat{\Psi}_h(B) = f_h(\hat{\Phi})$. Connaissant la matrice de variance covariance des coefficients du VAR , $\Gamma_{\hat{\Phi}}(0)$, la méthode du delta¹¹ permet de trouver celle des coefficients $\hat{\Psi}(B)$, soit $\Gamma_{\hat{\Psi}_h}(0) = F_h \Gamma_{\hat{\Phi}}(0) F_h'$ où $F_h = \frac{\partial f_h(\cdot)}{\partial \Phi|_{\phi = \hat{\phi}}}$ ¹². Les expressions analytiques de ces variances asymptotiques

11. Rappel : Soit X un vecteur de d variables aléatoires d'espérance μ_X , de variance $\Gamma_X(0)$. On pose $Y = f(X)$ où f est une fonction continue dérivable. Alors un développement de Taylor à l'ordre 1 au voisinage de μ_X donne : $Y = f(X) \approx f(\mu_X) + F(X - \mu_X)$, $F(\cdot)$ étant la dérivée partielle de $f(\cdot)$ évaluée en μ_X . Ainsi $\Gamma_Y(0) = F \Gamma_X(0) F'$.

12. Dans le cadre d'un VAR , les fonctions de passage f_h des coefficients autorégressifs Φ vers les coefficients Ψ du VMA sont relativement complexes et en tout cas non linéaires. On peut illustrer les manipulations dans le cadre simple d'un $AR(1)$ univarié stationnaire : $X_t = \phi_1 X_{t-1} + \varepsilon_t = \sum_{j=0}^{+\infty} \psi_j \varepsilon_{t-j}$ avec X_t une variable aléatoire réelle, $|\phi_1| < 1$, $\phi_0 = 1$. La réponse de X_{t+h} à un choc ε_t est donc donnée par $\phi_{j+h} = \phi_1^h$. Dans ce cadre, $F(\cdot) = h\phi_1^{h-1}$ est un scalaire, et finalement $V(\hat{\psi}_{i+h}) = (h\phi_1^{h-1})^2 V(\hat{\phi}_1)$, où $V(\hat{\phi}_1)$ est donnée par l'estimation de

sont cependant compliquées et cela explique en partie la popularité de la méthode bootstrap de construction des intervalles : celle-ci se passe en effet de la connaissance des expressions en question. L'autre argument est que sur petits échantillons les propriétés asymptotiques ne sont pas vérifiées et on opère alors que le taux de couverture des intervalles de confiance obtenus par bootstrap améliore celui des intervalles de confiance associés à la méthode delta.

3.2.4.2 Intervalles de confiance obtenus via la méthode bootstrap

On sait que si le $VAR(p)$ est bien spécifié, alors ε_t est un processus en bruit blanc : ses composantes $\varepsilon_{1,t}, \varepsilon_{2,t}, \dots, \varepsilon_{d,t}$ ne sont pas nécessairement orthogonales mais, en revanche, $\forall s \neq 0$, ε_t et ε_{t+s} le sont. Soit $\hat{\varepsilon} = (\hat{\varepsilon}'_1, \hat{\varepsilon}'_2, \dots, \hat{\varepsilon}'_T)$ la matrice de taille $(T \times d)$ dont la $i^{ème}$ ligne est constituée par le vecteur des résidus estimés au temps i . Chacun de ces vecteurs peut être vu comme la réalisation d'un tirage aléatoire équiprobable de ε_i . Le fait qu'il se soit réalisé au temps i n'est pas essentiel pour les propriétés des estimateurs du VAR puisqu'il avait autant de chance de se réaliser à un autre temps j . La méthode bootstrap exploite cette idée comme suit :

1. Estimation du VAR , récupération des coefficients $\hat{\phi}_j$, $j = 1, 2, \dots, p$ et des fonctions de réponse $\hat{\Psi}_{i,j}^h$ et/ou $\hat{\xi}_{i,j}^h$, construction de $\hat{\varepsilon}$.
2. Construction d'une matrice $\tilde{\varepsilon}$ de dimension $(T \times d)$ par tirage aléatoire avec remise dans les lignes de $\hat{\varepsilon}$.
3. Les p premières observations de X_t étant considérées comme données¹³, construction de d séries \tilde{X}_t selon :

$$\tilde{X}_t = \begin{cases} X_t & \text{pour } t = -p+1, -p+2, \dots, 0 \quad (\text{conditions initiales}) \\ \hat{c} + \hat{\Phi}_1 \tilde{X}_{t-1} + \dots + \hat{\Phi}_p \tilde{X}_{t-p} + \tilde{\varepsilon}'_t, & \text{pour } t = 1, 2, \dots, T \end{cases} \quad (3.43)$$

où $\tilde{\varepsilon}_t$ est la $t^{ème}$ ligne de $\tilde{\varepsilon}$.

4. Estimation du $VAR(p)$ au moyen des nouvelles séries \tilde{X}_t et des fonctions de réponse associées $\tilde{\psi}_{i,j}^h$ et/ou $\tilde{\xi}_{i,j}^h$.
5. Les étapes 2 à 4 sont répétées n fois. Au final on dispose, pour tout h , de n valeurs $\tilde{\psi}_{i,j}^h$ et/ou $\tilde{\xi}_{i,j}^h$. Un encadrement à, par exemple, 90% de confiance des $\hat{\psi}_{i,j}^h$ ou $\hat{\xi}_{i,j}^h$ obtenus à l'étape 1 est construit en prenant les 5^{ème} et 95^{ème} centiles obtenus sur ces n valeurs $\tilde{\psi}_{i,j}^h$ ou $\tilde{\xi}_{i,j}^h$, selon que l'on veut un IC sur les fonctions simples ou orthogonalisées.

Avec cette méthode, aucune hypothèse de distribution n'est imposée sur les innovations, si ce n'est leur indépendance dans le temps. C'est donc une procédure non paramétrique de construction d'estimateurs de bornes d'intervalles de confiance. Naturellement, si l'hypothèse d'indépendance n'est pas vérifiée, le tirage aléatoire avec remise dans les lignes de $\hat{\varepsilon}$ fait que la méthode de bootstrapping est déficiente puisque $\tilde{\varepsilon}$ ne peut plus être considérée comme une réalisation d'aléatoires de même loi que les innovations initiales.

Cependant, on notera que cette procédure simple à mettre en oeuvre donne des intervalles de confiance ayant un taux de couverture de la vraie fonction de réponse pouvant être très éloigné du taux nominal $(1 - \alpha)$. Pour certains, l'origine du problème est que le biais d'estimation sur les Φ_j , $j = 1, \dots, p$ sont amplifiés sur les $\tilde{\psi}_{i,j}^h$. Afin de corriger ces biais, une proposition relativement populaire a été avancée par Kilian (1998) (ou Kilian (1999)) : cette technique, dit

l' $AR(1)$.

13. Notons que dans certaines implémentations, le vecteur des conditions initiales est lui-même constitué d'une séquence de $p+1$ observations tirée au hasard parmi les $T+p+1$ valeurs observées. Cette pratique est dangereuse si le système est non stationnaire, ou proche de la non stationnarité, ces tirages successifs ne décrivant alors pas bien la distribution non conditionnelle des conditions initiales.

de *bootstrap after bootstrap*, consiste à effectuer une première fois les étapes 1 à 5 précédentes pour récupérer un ensemble, de n matrices de coefficients autorégressifs sur les échantillons bootstrappés, $\widehat{\Phi}_k^* = \widehat{\Phi}_{1,k}^*, \widehat{\Phi}_{2,k}^*, \dots, \widehat{\Phi}_{p,k}^*, k = 1, 2, \dots, n$. Si $\widehat{\Phi}$ est la matrice des coefficients obtenus sur l'échantillon initial, on estime le biais par $\widehat{\Phi} - \overline{\Phi}^*$, où $\overline{\Phi}^*$ est la moyenne des $\widehat{\Phi}_k^*$, pour construire un estimateur des coefficients autorégressifs corrigés du biais :

$$\widehat{\widehat{\Phi}} = \widehat{\Phi} + (\widehat{\Phi} - \overline{\Phi}^*) = 2\widehat{\Phi} - \overline{\Phi}^* \quad (3.44)$$

Les 5 étapes précédentes sont alors reprises une deuxième fois en remplaçant $\widehat{\Phi}$ par $\widehat{\widehat{\Phi}}$ dans la troisième.

Exemple 3.6 *Estimation de l'intervalle de confiance des fonctions de réponse impulsionnelle par Monte Carlo*

La méthode `res.irf_errband_mc()` les intervalles de confiances des fonctions de réponses obtenues par simulation de Monte Carlo et retourne les bornes inférieures et supérieures.

```
# Simulation Monte Carlo des bandes d'erreurs
mc = results.irf_errband_mc(orth=False, repl=1000, steps=4, signif=0.05,
                           seed=np.random.seed(123), burn=100, cum=False)
v1 = pd.DataFrame(np.append(mc[0][0], np.append(mc[0][1], mc[0][2], axis=0), axis=0),
                  columns = ["dln_inv.lo", "dln_inc.lo", "dln_consump.lo"])
v2 = pd.DataFrame(np.append(mc[1][0], np.append(mc[1][1], mc[1][2], axis=0), axis=0),
                  columns = ["dln_inv.up", "dln_inc.up", "dln_consump.up"])
mc_value = pd.concat([v1, v2], axis=1); mc_value = mc_value.iloc[:, [0, 3, 1, 4, 2, 5]]
mc_value.index = [x for x in range(9)]
```

Table 3.3 – Intervalles de confiance obtenus par Monte Carlo

	Intervalle sur dln_inv		Intervalle sur dln_inc		Intervalle sur dln_consump	
	dln_inv.lo	dln_inv.up	dln_inc.lo	dln_inc.up	dln_consump.lo	dln_consump.up
h = 1						
0	1.0000	1.0000	0.0000	0.0000	0.0000	0.0000
1	0.0000	0.0000	1.0000	1.0000	0.0000	0.0000
2	0.0000	0.0000	0.0000	0.0000	1.0000	1.0000
h = 2						
3	-0.5609	-0.0652	-0.9614	1.3501	-0.4552	2.2302
4	-0.0178	0.1074	-0.4471	0.1385	-0.0546	0.6004
5	-0.0523	0.0499	0.0090	0.4764	-0.5488	-0.0206
h = 3						
6	-0.2901	0.1787	-0.8004	1.3603	-0.9124	1.7193
7	-0.0312	0.0861	-0.1470	0.3807	-0.4077	0.1869
8	-0.0065	0.0915	0.0513	0.4945	-0.1630	0.3449

3.2.5 Décomposition de la variance des erreurs de prévisions

La prévision à un horizon h de X est donnée par $X_{t+h|t} = \mathbb{E}_t[X_{t+h}]$. En conséquence, l'expression de l'erreur associée est, si on utilise l'écriture du VMA sur les chocs orthogonaux, équation (3.36),

$$e_{t+h|t} = X_{t+h} - X_{t+h|t} = \Xi_0\omega_{t+h} + \Xi_1\omega_{t+h-1} + \dots + \Xi_{h-1}\omega_{t+1} \quad (3.45)$$

Fort logiquement, les erreurs de prévision à un horizon h sont déterminées par les innovations qui se produiront au cours des h périodes à venir. On a évidemment :

$$\mathbb{E} [e_{t+h|t} | I_t] = 0 \quad (3.46)$$

L'expression de l'erreur afférente à une des variables X_i , $1 \leq i \leq d$ quelconque est ainsi :

$$e_{i,t+h|t} = X_{i,t+h} - X_{i,t+h|t} \quad (3.47)$$

$$= \sum_{j=1}^{j=d} \xi_{i,j}^0 \omega_{j,t+h} + \sum_{j=1}^{j=d} \xi_{i,j}^1 \omega_{j,t+h-1} + \dots + \sum_{j=1}^{j=d} \xi_{i,j}^{h-1} \omega_{j,t+1} \quad (3.48)$$

Sous cette forme, on pourrait mesurer la contribution de l'ensemble des innovations se produisant à un temps $t+l$ à la variance de l'erreur à l'horizon h , avec $1 \leq l \leq h$.

Si on désire faire apparaître la contribution à cette variance de chacune des innovations ω_j , $1 \leq j \leq d$, il suffit de réécrire l'expression précédente en les isolants, soit :

$$e_{i,t+h|t} = \sum_{l=1}^{l=h} \xi_{i,1}^{h-l} \omega_{1,t+l} + \sum_{l=1}^h \xi_{i,2}^{h-l} \omega_{2,t+l} + \dots + \sum_{l=1}^h \xi_{i,d}^{h-l} \omega_{d,t+l} \quad (3.49)$$

Compte-tenu du fait que ω_t est un processus en bruit blanc constitué de variables aléatoires de variance unitaire et orthogonales entre elles, il vient alors :

$$\text{Var} (e_{i,t+h|t}) = \mathbb{E} [e_{i,t+h|t}^2] = \sum_{l=1}^{l=h} (\xi_{i,1}^{h-l})^2 + \sum_{l=1}^{l=h} (\xi_{i,2}^{h-l})^2 + \dots + \sum_{l=1}^{l=h} (\xi_{i,d}^{h-l})^2 \quad (3.50)$$

et donc $\sum_{l=1}^{l=h} (\xi_{i,j}^{h-l})^2$, avec $1 \leq j \leq d$ est la contribution du choc ω_j à la variance de $e_{i,t+h|t}$.

En termes de proportions, ce qui a l'avantage de révéler rapidement quels chocs contribuent le plus à cette variance, la proportion de $\text{Var} (e_{i,t+h|t})$ attribuable à $\omega_j, j = 1, 2, \dots, d$ est donnée par :

$$\frac{\sum_{l=1}^{l=h} (\xi_{i,j}^{h-l})^2}{\sum_{l=1}^{l=h} (\xi_{i,1}^{h-l})^2 + \sum_{l=1}^{l=h} (\xi_{i,2}^{h-l})^2 + \dots + \sum_{l=1}^{l=h} (\xi_{i,d}^{h-l})^2} \quad (3.51)$$

Les valeurs estimées sont obtenues en remplaçant les termes $(\xi_{i,j}^{h-l})^2$ par leurs estimations $(\hat{\xi}_{i,j}^{h-l})^2$ dans l'équation précédente. Enfin, la procédure de bootstrapping décrite lors de la construction d'IC sur les fonctions de réponse peut également être employée pour le calcul d'intervalles de confiance sur ces proportions.

Exemple 3.7 *Décomposition de la variance de l'erreur*

```
# Décomposition de la variance de l'erreur
fevd = results.fevd(5)
fevd.summary()

## FEVD for dln_inv
##      dln_inv  dln_inc  dln_consump
## 0  1.000000  0.000000  0.000000
## 1  0.959959  0.017511  0.022530
## 2  0.945648  0.028021  0.026330
## 3  0.940791  0.029361  0.029847
## 4  0.938463  0.030181  0.031356
##
## FEVD for dln_inc
##      dln_inv  dln_inc  dln_consump
## 0  0.017536  0.982464  0.000000
## 1  0.060245  0.907471  0.032284
## 2  0.069592  0.895764  0.034644
## 3  0.068312  0.892322  0.039365
## 4  0.068500  0.892121  0.039378
##
## FEVD for dln_consump
##      dln_inv  dln_inc  dln_consump
## 0  0.079950  0.272922  0.647128
## 1  0.077247  0.273849  0.648904
## 2  0.129729  0.333642  0.536629
## 3  0.128704  0.334988  0.536308
## 4  0.128588  0.339245  0.532166

# FEVD plot
results.fevd(5).plot();
plt.show()
```

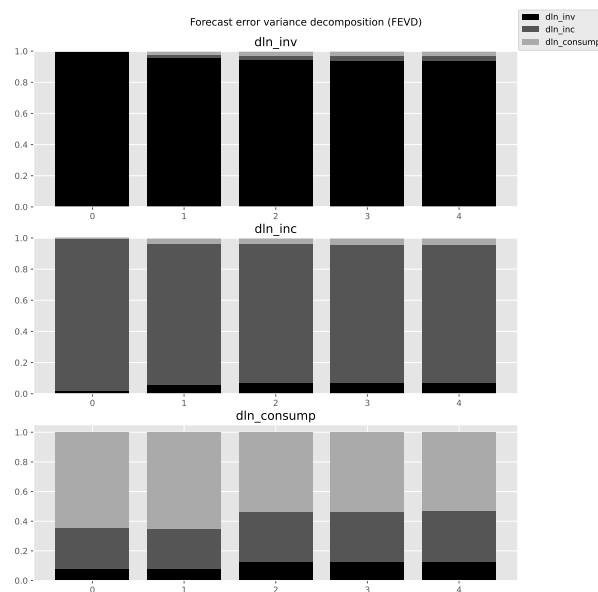


Figure 3.5 – Décomposition de la variance de l'erreur de prévision

Deuxième partie

Économétrie des processus non stationnaires

Cointégration et modèle à correction d'erreur

Sommaire

4.1 La notion de cointégration	89
4.2 Le modèle à correction d'erreur	91

La théorie de la cointégration permet d'étudier des séries non stationnaires mais dont une combinaison linéaire est stationnaire. Elle permet ainsi de spécifier des relations stables à long terme tout en analysant conjointement la dynamique de court terme des variables considérées.

4.1 La notion de cointégration

4.1.1 Rappels sur les tests de racine unitaire

4.1.2 Justification de la cointégration

Le point de départ de la cointégration réside dans le fait que de nombreuses séries macroéconomiques et financières sont non stationnaires. Or, si l'on applique les méthodes habituelles, deux principaux problèmes surgissent :

1. Le problème des régressions fallacieuses mis en avant par Granger and Newbold (1974). En effet, soient $X_{1,t}$ et $X_{2,t}$ deux séries intégrées d'ordre 1 suivant chacune une marche aléatoire sans dérive :

$$\begin{cases} X_{1,t} = X_{1,t-1} + \varepsilon_{1,t} \\ X_{2,t} = X_{2,t-1} + \varepsilon_{2,t} \end{cases} \quad (4.1)$$

$\varepsilon_{1,t}$ et $\varepsilon_{2,t}$ sont deux bruits blancs indépendants. Si l'on effectue la régression $X_{2,t} = \alpha_0 + \alpha_1 X_{1,t} + \eta_t$, on devrait avoir $\alpha_1 = 0$. Or Granger and Newbold (1974), en effectuant des simulations de Monte Carlo, montrent qu'en fait α_1 est significativement différent de zéro. En d'autres termes, $X_{1,t}$ apparaît comme une variable explicative de $X_{2,t}$, ce qui n'a aucun sens puisque, par hypothèse, les deux séries sont indépendantes. La non stationnarité a ainsi pour conséquence que les procédures d'inférence classiques ne sont plus valables. En effet, si $X_{2,t}$ est $I(1)$, cela signifie que, sous l'hypothèse nulle, $\varepsilon_{2,t}$ sera aussi $I(1)$. Or, on sait que les tests usuels de significativité des coefficients (tests de Fisher ou de Student) reposent sur l'hypothèse de bruit blanc des résidus, hypothèse qui n'est pas vérifiée ici.

2. Certaines lois asymptotiques ne sont plus valables. Par exemple, les statistiques des tests de Dickey - Fuller ne suivent plus une loi habituelle.

La modélisation ARMA n'est valable que pour des séries stationnaires. Si les séries ne sont pas stationnaires stochastiquement, on a vu que les différencier suffisait à les rendre stationnaires. Cette opération de différenciation a cependant deux limites principales :

1. Elle ne prend en compte que les changements intervenus d'une période sur l'autre. Les relations entre les niveaux des variables ne sont pas étudiées. On masque alors les propriétés de long terme des séries.
2. Si des relations stables existent à long terme entre des variables, on parle de relations de cointégration. Les modèles classiques, tels que les processus ARMA, sont incapables d'en rendre compte. Pour cela, il faut utiliser une classe particulière de modèles, directement liés à la théorie de la cointégration : les modèles à correction d'erreur.

4.1.3 Définition de la cointégration

La théorie de la cointégration a été introduite par Granger (1981) et a connu depuis de très nombreux développements. Le lien entre cointégration et modèles à correction d'erreur est explicité dans Granger and Weiss (1983) et Engle and Granger (1987).

Définition 4.1 Série intégrée d'ordre d

Une série est intégrée d'ordre d (notée $X_t \sim I(d)$), s'il convient de la différencier d fois afin de la stationnariser.

La série (X_t) sera dite intégrée d'ordre d ($d \geq 1$) si $\Delta^{d-1}X_t$ n'est pas stationnaire et $\Delta^d X_t$ est stationnaire. Une série stationnaire sera dite intégrée d'ordre 0.

Propriété 4.1 Propriétés des séries intégrées

De façon générale, on peut mettre en avant quatre grandes propriétés aux séries intégrées :

1. Si $X_t \sim I(d)$, alors $\alpha_0 + \alpha_1 X_t \sim I(d)$ où α_0 et α_1 sont des constantes, avec $\beta \neq 0$.
2. Si $X_{1,t} \sim I(0)$ et $X_{2,t} \sim I(0)$, alors $\alpha_1 X_{1,t} + \alpha_2 X_{2,t} \sim I(0)$ où α_1 et α_2 sont des constantes
3. Si $X_{1,t} \sim I(0)$ et $X_{2,t} \sim I(1)$ alors $\alpha_1 X_{1,t} + \alpha_2 X_{2,t} \sim I(1)$ où α_1 et α_2 sont des constantes avec $\alpha_2 \neq 0$ ($I(1)$ est une propriété dominante).
4. Si $X_{1,t} \sim I(d_1)$ et $X_{2,t} \sim I(d_2)$ alors en général $\alpha_1 X_{1,t} + \alpha_2 X_{2,t} \sim I(\max(d_1, d_2))$ où α_1 et α_2 sont des constantes non nulles.

Soit deux séries $X_{1,t}$ et $X_{2,t}$ intégrées d'ordre d , on se pose la question de savoir : quel est l'ordre d'intégration de la série Z_t définie par :

$$Z_t = \alpha_1 X_{1,t} + \alpha_2 X_{2,t} \quad (4.2)$$

avec α_1 et α_2 deux constantes non nulles.

En effet, le résultat dépend des signes des coefficients α_1 et α_2 et de l'existence d'une dynamique non stationnaire commune.

Définition 4.2 Cointégration

Si x_t et y_t sont deux séries non stationnaires et toutes intégrées d'ordre d , alors en général la combinaison linéaire z_t définie par :

$$z_t = x_t - \alpha y_t \quad (4.3)$$

est aussi $I(d)$.

Cependant, il est possible que z_t ne soit pas $I(d)$ mais $I(d - b)$ où b est un entier positif. Dans ce cas, x_t et y_t sont dites cointégrées. α est le paramètre de cointégration et le vecteur $[1, -\alpha]$ est le vecteur de cointégration.

Le cas le plus étudié correspond à : $d = b = 1$. Ainsi, deux séries non stationnaires ($I(1)$) sont cointégrées s'il existe une combinaison linéaire stationnaire ($I(0)$) de ces deux séries.

Propriété 4.2 Conditions de cointégration

Deux séries x_t et y_t sont dites cointégrées si les deux conditions suivantes sont vérifiées :

- Elles sont affectées d'une tendance stochastique de même ordre d'intégration d .
- Une combinaison linéaire de ces séries permet de se ramener à une série d'ordre d'intégration inférieure.

Soit $x_t \sim I(d)$ et $y_t \sim I(d)$ tel que $\alpha x_t + \beta y_t \sim I(d - b)$ avec $d \geq b \geq 0$. On dit que x_t et y_t sont cointégrées et on note $x_t, y_t \sim CI(d, b)$ et $[\alpha, \beta]$ est appelé vecteur de cointégration.

4.2 Le modèle à correction d'erreur

Le modèle à correction d'erreur (*Error Correction Model*) introduit notamment par Hendry, Pagan, and Sargan (1984) permet de modéliser les ajustements qui conduisent à une situation d'équilibre de long terme. Il s'agit d'un modèle dynamique qui intègre à la fois les évolutions de court terme et de long terme des variables.

4.2.1 Estimation du modèle à correction d'erreur

4.2.1.1 Méthode d'estimation de Engle et Granger

La méthode d'estimation d'un modèle à correction d'erreur a été proposée par Engle and Granger (1987) et son principale avantage réside dans sa simplicité de mise en oeuvre. Elle n'est valable que pour les séries $CI(1, 1)$.

Soient x_t et y_t deux séries.

— Etape 1 : Tester l'ordre d'intégration des variables

Une condition nécessaire de cointégration est que les séries doivent être intégrées de même ordre. Si les séries ne sont pas intégrées de même ordre, elles ne peuvent pas être cointégrées. Par conséquent, si les séries étudiées ne sont pas intégrées de même ordre, la procédure est arrêtée, il n'y a pas de risque de cointégration.

— Etape 2 : Estimation de la relation de long terme

Si la condition nécessaire est vérifiée, on estime par les MCO la relation de long terme entre les variables :

$$y_t = \alpha_0 + \alpha_1 x_t + \varepsilon_t \quad (4.4)$$

où ε_t est le terme d'erreur. L'estimateur des moindres carrés ordinaires converge vers sa vraie valeur si les variables sont cointégrées.

— **Etape 3 : Tester la stationnarité des résidus**

Pour que la relation de cointégration soit acceptée, le résidu $\hat{\varepsilon}_t$, issu de la relation de long terme doit être stationnaire :

$$\hat{\varepsilon}_t = y_t - \hat{\alpha}_0 - \hat{\alpha}_1 x_t \quad (4.5)$$

La stationnarité du résidu est testée à l'aide des tests de Dickey - Fuller simple ou de Dickey - Fuller augmenté. Si le résidu est stationnaire, nous pouvons alors estimer le modèle à correction d'erreur.

— **Etape 4 : Estimation du modèle à correction d'erreur**

On estime le modèle à correction d'erreur par les MCO :

$$\Delta y_t = \phi \Delta x_t + \theta \hat{\varepsilon}_{t-1} + \eta_t \quad \theta < 0 \quad (4.6)$$

où $\eta_t \sim BB$ et $\hat{\varepsilon}_t$ est le résidu estimé de la relation de long terme (equation (4.4)) retardé d'une période : $\hat{\varepsilon}_{t-1} = y_{t-1} - \hat{\alpha}_0 - \hat{\alpha}_1 x_{t-1}$. Le coefficient θ , appelé force de rappel vers l'équilibre, doit être significativement négatif ; dans le cas contraire, il convient de rejeter une spécification de type ECM.

Engle and Granger (1987) ont montré que si le vecteur de cointégration $(\hat{\alpha}_0, \hat{\alpha}_1)$ a été correctement estimé dans la deuxième étape, alors :

- Les estimateurs $\hat{\theta}$ et $\hat{\phi}$ sont convergents et équivalents à ceux du maximum de vraisemblance.
- Les écarts types des coefficients estimés par les MCO sont des estimateurs convergents des vrais écarts types.

Les résultats énoncés ci-dessus montrent que l'on peut utiliser, dans la quatrième étape, les tests usuels (de type Fisher ou Student) pour effectuer des tests de significativité des paramètres caractérisant la dynamique de court terme (ceci provient du fait que tous les termes intervenant dans l'équation (4.6) sont $I(0)$).

Au cours de la première étape de la méthode d'estimation de Engle and Granger (1987), il est nécessaire de vérifier que les séries x_t et y_t sont bien cointégrées, c'est - à - dire que les résidus de la relation de long terme sont bien stationnaires. Pour cela, on applique des tests de cointégration.

Proposition 4.1

De façon générale que (4.6), le modèle à correction d'erreur peut s'écrire comme suit :

$$\Delta y_t = c + \sum_{i=1}^{i=p} a_i \Delta y_{t-i} + \sum_{j=0}^{j=q} b_j \Delta x_{t-j} + \theta (y_{t-1} - \hat{\alpha}_0 - \hat{\alpha}_1 x_{t-1}) + \eta_t \quad (4.7)$$

où les variables interviennent soit à travers leurs différences premières (supposées déterministes), soit à travers un terme d'écart à la cible à long terme, à la période précédente.

4.2.1.2 Tests de cointégration

Engle and Granger (1987) ont proposé divers tests afin d'appréhender l'hypothèse de cointégration. On peut citer :

- Le test CRDW (Cointegration Regression Durbin Watson)
- Les tests de Dickey - Fuller (DF) et DF augmenté (ADF)
- Les tests RVAR (Restricted VAR) et ARVAR (Augmented RVAR)
- Tests UVAR (Unrestricted VAR) et AUVAR (Augmented UVAR)

Dans tous les cas, l'hypothèse nulle est celle d'absence de cointégration, l'hypothèse alternative étant la cointégration. On présente ici la procédure du test DF et ADF. Il a pour objet de tester l'existence d'une racine unitaire dans les résidus estimés $\hat{\varepsilon}_t$ de la relation de long terme.

$$\hat{\varepsilon}_t = y_t - \hat{\alpha}_0 - \hat{\alpha}_1 x_t \quad (4.8)$$

On teste l'hypothèse nulle $H_0 : \hat{\varepsilon}_t$ non stationnaire traduisant le fait que les séries x_t et y_t sont non cointégrées, contre l'hypothèse alternative $H_1 : \hat{\varepsilon}_t$ stationnaire indiquant que les séries x_t et y_t sont cointégrées.

Remarque 4.1

Le modèle à correction d'erreur permet de modéliser conjointement les dynamiques de court terme (représentées par les variables en différence première) et de long terme (représentées par les variables en niveau). La dynamique de court terme s'écrit :

$$y_t = a_0 + a_1 y_{t-1} + a_2 x_t + a_3 x_{t-1} + \mu_t, \quad a_1 < 1 \quad (4.9)$$

La dynamique de long terme s'exprime de la manière suivante :

$$y_t = \alpha_0 + \alpha_1 x_t + \varepsilon_t \quad (4.10)$$

car à long terme, on a $y_{t-1} = y_t$ et $x_{t-1} = x_t$ et la dynamique de court terme devient à long terme :

$$\begin{aligned} y_t &= a_0 + a_1 y_t + a_2 x_t + a_3 x_t + \mu_t \\ (1 - a_1) y_t &= a_0 + (a_2 + a_3) x_t + \mu_t \\ y_t &= \frac{a_0}{1 - a_1} + \frac{a_2 + a_3}{1 - a_1} x_t + \frac{1}{1 - a_1} \mu_t \\ y_t &= \alpha_0 + \alpha_1 x_t + \varepsilon_t \end{aligned}$$

avec $\alpha_0 = \frac{a_0}{1 - a_1}$, $\alpha_1 = \frac{a_2 + a_3}{1 - a_1}$, $\varepsilon_t = \frac{\mu_t}{1 - a_1}$ et $a_1 < 1$.

Le modèle à correction d'erreur s'obtient à partir de la dynamique de court terme. En effet, on a :

$$\begin{aligned} y_t &= a_0 + a_1 y_{t-1} + a_2 x_t + a_3 x_{t-1} + \mu_t \\ y_t - y_{t-1} &= a_0 + a_1 y_{t-1} - y_{t-1} + a_2 x_t - a_2 x_{t-1} + a_2 x_{t-1} + a_3 x_{t-1} + \mu_t \\ \Delta y_t &= a_0 - (1 - a_1) y_{t-1} + a_2 (x_t - x_{t-1}) + (a_2 + a_3) x_{t-1} + \mu_t \\ \Delta y_t &= -(1 - a_1) \left(y_{t-1} - \frac{a_2 + a_3}{1 - a_1} x_{t-1} - \frac{a_0}{1 - a_1} \right) + a_2 \Delta x_t + \mu_t \\ \Delta y_t &= \phi \Delta x_t + \theta (y_{t-1} - \alpha_0 - \alpha_1 x_{t-1}) + \mu_t \end{aligned}$$

avec $\phi = a_2, \theta = -(1 - a_1), \alpha_1 = \frac{a_2 + a_3}{1 - a_1}, \alpha_0 = \frac{a_0}{1 - a_1}$ et $a_1 < 1$.

Exemple 4.1 *Etude de l'évolution de la consommation et du revenu de FONGO - NDENG*

Considérons les séries mensuelles de consommation et de revenu disponible brut des ménages de FONGO - NDENG allant de janvier 2018 à juin 2022. Ces séries sont notées respectivement y_t et x_t ¹.

La figure 4.1 indique que les deux séries semblent exhiber une tendance commune à la hausse. Ces deux séries paraissent non stationnaires et, du fait de leur apparente évolution similaire, il est légitime de s'intéresser à l'étude de la cointégration entre les deux variables. On demande d'estimer la relation entre ces deux variables en testant une éventuelle cointégration (dans ce cas estimer le modèle à correction d'erreur).

```
# Chargement des données
import pandas as pd; import matplotlib.pyplot as plt
eg_df = pd.read_excel("./donnee/engle_granger_ecm.xls", index_col=0)
eg_df.index = pd.date_range('2018-01-01', periods=len(eg_df), freq = "M")

# Représentation graphique
fig, axe = plt.subplots(figsize=(16,6))
axe.plot(eg_df.X, color = "black", label = "$x_{t}$");
axe.plot(eg_df.Y, color = "black", linestyle = "--", label = "$y_{t}$");
axe.set_xlabel("mois");
axe.legend();
plt.show()
```

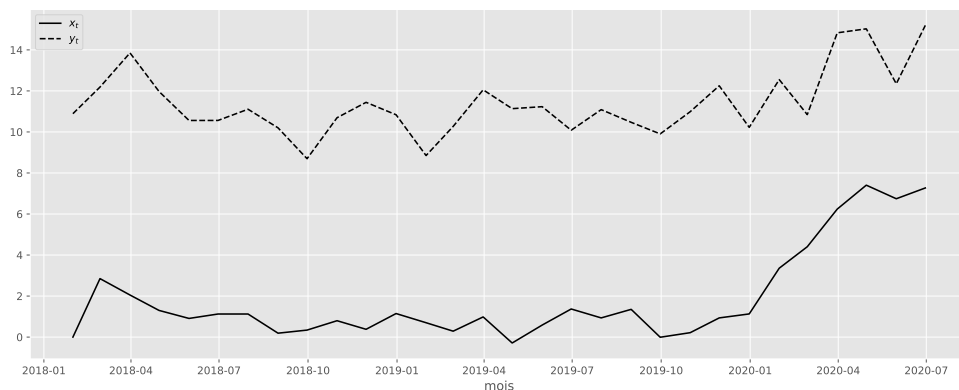


Figure 4.1 – Evolution des séries x_t et y_t

Dans la mesure où la méthode de Engle and Granger (1987) n'est valable que pour des séries intégrées d'ordre 1, il convient tout d'abord de déterminer l'ordre d'intégration de chacune des séries.

— **Etape 1** : Détermination de l'ordre d'intégration des séries x_t et y_t

La première étape consiste à étudier les propriétés des deux séries en termes de stationnarité. On adopte une stratégie à la Dickey - Fuller en appliquant le test de Dickey - Fuller simple sur les séries x_t et y_t .

1. Ces données ont été simulées pour un besoin d'étude et peuvent ne pas refléter la réalité.

— Test DF sur la série x_t

On part du modèle [3] : $\Delta x_t = \rho x_{t-1} + c + bt + \varepsilon_t$

```
# Fonction extractParams
import numpy as np
from mizani.formatters import scientific_format
def extractParams(res, model_type = "lm", seuil = 0.05, approx = 4):
    stderr = list()
    if model_type == "lm" or model_type == "arima":
        stderr = np.sqrt(np.diag(res.cov_params()))
    elif model_type == "arch":
        stderr = np.sqrt(np.diag(res.param_cov))
    level_inf = seuil/2; level_sup = 1 - level_inf
    params = pd.DataFrame({
        "coef" : res.params, "std err" : stderr, "t" : res.tvalues,
        "P>|t|" : scientific_format()(res.pvalues),
        "["+str(level_inf) : list(res.conf_int(alpha = seuil).values[:,0]),
        str(level_sup)+"]" : list(res.conf_int(alpha = seuil).values[:,1]),
        }, index = list(res.params.index)).round(approx)
    return params

# Dickey - Fuller : Modèle [3]
from arch.unitroot import ADF
resx_df3 = ADF(eg_df.X, lags=0, trend = 'ct').regression
resx_df3_params = extractParams(resx_df3, model_type = "lm")
```

Table 4.1 – Coefficients du modèle [3] du test DF sur X

	coef	std err	t	P> t	[0.025	0.975]
Level.L1	-0.0902	0.1093	-0.8253	4.167e-01	-0.3149	0.1345
const	-0.1672	0.3800	-0.4400	6.636e-01	-0.9484	0.6140
trend	0.0379	0.0262	1.4456	1.602e-01	-0.0160	0.0919

Table 4.2 – Valeurs critiques du modèle [3] pour X

	statistic	1pct	5pct	10pct
tau3	-0.8253	-4.15	-3.50	-3.18
phi2	1.3125	7.02	5.13	4.31
phi3	1.0462	9.31	6.73	5.61

La valeur empirique $t_{\hat{\rho}} = -0.8253 >$ aux trois valeurs critiques, on accepte l'hypothèse H_0 . Nous effectuons maintenant le test de l'hypothèse $H_0^3 : (c; b; \rho) = (c; 0; 1)$. La statistique $F_3 = 1.0462 <$ aux trois valeurs critiques pour Φ_3 . Nous sommes amenés à accepter les hypothèses H_0 et H_0^3 . Nous effectuons le test de l'hypothèse H_0^2 , soit le calcul de la statistique F_2 . Nous sommes amenés à accepter l'hypothèse H_0^2 , car $F_2 = 1.3125 <$ aux trois valeurs critiques pour Φ_2 .

Nous estimons maintenant le modèle [2] : $\Delta x_t = \rho x_{t-1} + c + \varepsilon_t$

```
# Dickey - Fuller : Modèle [2]
resx_df2 = ADF(eg_df.X, lags=0, trend = 'c').regression
resx_df2_params = extractParams(resx_df2, model_type = "lm")
```

Table 4.3 – Coefficients du modèle [2] du test DF sur X

	coef	std err	t	P> t	[0.025	0.975]
Level.L1	-0.0046	0.0937	-0.0491	9.612e-01	-0.1969	0.1877
const	0.2584	0.2451	1.0541	3.012e-01	-0.2446	0.7613

Table 4.4 – Valeurs critiques du modèle [2] pour X

	statistic	1pct	5pct	10pct
tau2	-0.0491	-3.58	-2.93	-2.60
phi1	0.8880	7.06	4.86	3.94

La valeur empirique $t_{\hat{\rho}} = -0.0491 >$ aux trois valeurs critiques, on accepte l'hypothèse H_0 . Nous effectuons maintenant le test de l'hypothèse H_0^1 . La statistique $F_1 = 0.8880 <$ aux trois valeurs critiques pour Φ_1 . Nous sommes amenés à accepter les hypothèses H_0 et H_0^1 .

Nous effectuons le test de nullité de la moyenne : $H_0 : \mu = 0$.

```
# Test de nullité de la moyenne
import scipy.stats as st
xtstat = st.ttest_1samp(eg_df.X, popmean=0)
print("statistic : %.4f et pvalue : %.4f"%(xtstat[0], xtstat[1]))

## statistic : 4.5260 et pvalue : 0.0001
```

La moyenne est significativement différente de 0. Le processus est une marche aléatoire au hasard sans dérive.

— Test DF sur la série y_t

On part du modèle [3] : $\Delta y_t = \rho y_{t-1} + c + bt + \varepsilon_t$

```
# Dickey - Fuller : Modèle [3]
resy_df3 = ADF(eg_df.Y, lags=0, trend = 'ct').regression
resy_df3_params = extractParams(resy_df3, model_type = "lm")
```

Table 4.5 – Coefficients du modèle [3] du test DF sur Y

	coef	std err	t	P> t	[0.025	0.975]
Level.L1	-0.5972	0.1954	-3.0569	5.124e-03	-0.9988	-0.1956
const	6.0992	2.1392	2.8512	8.420e-03	1.7021	10.4964
trend	0.0524	0.0338	1.5495	1.333e-01	-0.0171	0.1219

Table 4.6 – Valeurs critiques du modèle [3] pour Y

	statistic	1pct	5pct	10pct
tau3	-3.0569	-4.15	-3.50	-3.18
phi2	3.3968	7.02	5.13	4.31
phi3	4.9466	9.31	6.73	5.61

La valeur empirique $t_{\hat{\rho}} = -3.0569 >$ aux trois valeurs critiques, on accepte l'hypothèse H_0 . Nous effectuons maintenant le test de l'hypothèse $H_0^3 : (c; b; \rho) = (c; 0; 1)$. La statistique $F_3 = 4.9466 <$ aux trois valeurs critiques pour Φ_3 . Nous sommes amenés à accepter les hypothèses H_0 et H_0^3 .

Nous effectuons le test de l'hypothèse H_0^2 , soit le calcul de la statistique F_2 . Nous sommes amenés à accepter l'hypothèse H_0^2 , car $F_2 = 3.3968 <$ aux trois valeurs critiques pour Φ_2 .

Nous estimons maintenant le modèle [2] : $\Delta y_t = \rho y_{t-1} + c + \varepsilon_t$

```
# Dickey - Fuller : Modèle [2]
resy_df2 = ADF(eg_df.Y,lags=0,trend = 'c').regression
resy_df2_params = extractParams(resy_df2, model_type = "lm")
```

Table 4.7 – Coefficients du modèle [2] du test DF sur Y

	coef	std err	t	P> t	[0.025	0.975]
Level.L1	-0.5143	0.1927	-2.6688	1.272e-02	-0.9097	-0.1189
const	5.9503	2.1918	2.7148	1.142e-02	1.4531	10.4474

Table 4.8 – Valeurs critiques du modèle [2] pour Y

	statistic	1pct	5pct	10pct
tau2	-2.6688	-3.58	-2.93	-2.60
phi1	3.7026	7.06	4.86	3.94

La valeur empirique $t_{\hat{\rho}} = -2.6688 >$ à la valeur critique au seuil de 5%, on accepte l'hypothèse H_0 . Nous effectuons maintenant le test de l'hypothèse H_0^1 . La statistique $F_1 = 3.7026 <$ aux trois valeurs critiques pour Φ_1 . Nous sommes amenés à accepter les hypothèses H_0 et H_0^1 .

Nous effectuons le test de nullité de la moyenne : $H_0 : \mu = 0$.

```
# Test de nullité de la moyenne
import scipy.stats as st
ytstat = st.ttest_1samp(eg_df.Y,popmean=0)
print("statistic : %.4f et pvalue : %.4f"%(ytstat[0], ytstat[1]))
```

```
## statistic : 38.6950 et pvalue : 0.0000
```

La moyenne est significativement différente de 0. Le processus est une marche aléatoire au hasard sans dérive.

Des tests similaires sur les différences premières de x_t et y_t indiquent qu'elles sont stationnaires, les deux séries étudiées sont $I(1)$, il existe donc un risque de cointégration.

— **Etape 2** : Estimation de la relation de long terme

On estime la relation de long terme suivante : $y_t = \alpha_0 + \alpha_1 x_t + \varepsilon_t$

```
# Estimation de la relation de long terme
import statsmodels.formula.api as smf
model = smf.ols('Y~X', data = eg_df).fit()
lm_params = extractParams(model, model_type = "lm")
```

```
# Coefficient de détermination
print("R2 score : %.4f"%(model.rsquared))
```

```
## R2 score : 0.5920
```

Table 4.9 – Estimation de la relation de long terme

	coef	std err	t	P> t	[0.025	0.975]
Intercept	10.3844	0.2504	41.4664	1.107e-26	9.8715	10.8974
X	0.5516	0.0865	6.3735	6.739e-07	0.3743	0.7289

Le tableau 4.9 donne les résultats de l'estimation de la relation statique entre y et x . Il va de soi que l'on explique la consommation par le revenu.

On déduit de l'estimation de la relation statique la série des résidus. On rappelle que Si les résidus sont non stationnaires, la relation estimée dans le tableau 4.9 est une régression fallacieuse. Si les résidus sont stationnaires, la relation estimée dans le tableau (4.9) est une relation de cointégration.

— **Etape 3** : Test DF sur les résidus de la relation de long terme

Nous vérifions que le résidu est bien stationnaire :

```
# Dickey - Fuller : Modèle [3]
resid = model.resid
resid_df3 = ADF(resid,lags=0,trend = 'ct').regression
resid_df3_params = extractParams(resid_df3, model_type = "lm")
```

Table 4.10 – Coefficients du modèle [3] du test DF sur $\hat{\varepsilon}_t$

	coef	std err	t	P> t	[0.025	0.975]
Level.L1	-1.0530	0.1997	-5.2720	1.642e-05	-1.4635	-0.6424
const	0.2204	0.4156	0.5304	6.004e-01	-0.6338	1.0747
trend	-0.0160	0.0244	-0.6548	5.184e-01	-0.0660	0.0341

Table 4.11 – Valeurs critiques du modèle [3] pour $\hat{\varepsilon}_t$

	statistic	1pct	5pct	10pct
tau3	-5.2720	-4.15	-3.50	-3.18
phi2	9.3198	7.02	5.13	4.31
phi3	13.9784	9.31	6.73	5.61

La valeur empirique $t_{\hat{\rho}} = -5.2720 <$ aux trois valeurs critiques, on rejette l'hypothèse H_0 . Nous effectuons maintenant le test de l'hypothèse H_0^3 . La statistique $F_3 = 13.9784 >$ aux trois valeurs critiques pour Φ_3 . Nous rejettons les hypothèses H_0 et H_0^3 . Nous effectuons ensuite le test de nullité du paramètre b : la statistique de test $t_{\hat{b}} = -0.6548$ admet une p-value égale à 0.5184 supérieure à 5%, on accepte l'hypothèse de nullité de b . Nous effectuons enfin le test de nullité de la constante c : la statistique de test $t_{\hat{c}} = 0.5304$ admet une p-value égale à 0.6004 supérieure à 5%, on accepte l'hypothèse de nullité de c . En conclusion, le résidu est bien stationnaire, il existe donc un risque de cointégration entre les deux variables.

— **Etape 4** : Estimation du modèle à correction d'erreur

Modélisons le taux de croissance de la consommation en fonction des résidus de la période précédente ($\widehat{\varepsilon}_{t-1}$), du taux de croissance du revenu. Nous calculons, d'abord, le résidu décalé d'une période, soit : $\hat{\varepsilon}_{t-1} = y_{t-1} - 10.38 - 0.55x_{t-1}$, puis nous estimons par les MCO le modèle : $\Delta y_t = \phi \Delta x_t + \theta \hat{\varepsilon}_{t-1} + \eta_t$.

```
# Estimation du modèle à correction d'erreur
df = pd.concat([eg_df.X.diff(),eg_df.Y.diff(),resid.shift(1)], axis = 1)
df.columns = ["X", "Y", "ecm"]
ecm = smf.ols('Y~X+ecm-1',data = df.dropna()).fit()
ecm_params = extractParams(ecm, model_type = "lm")
```

Table 4.12 – Estimation du modèle à correction d'erreur (ECM)

	coef	std err	t	P> t	[0.025	0.975]
X	0.6089	0.1972	3.0874	4.632e-03	0.2042	1.0135
ecm	-1.0189	0.1950	-5.2249	1.665e-05	-1.4191	-0.6188

```
# Coefficient de détermination
print("R2 score : %.4f"%(ecm.rsquared))
```

```
## R2 score : 0.6078
```

Le tableau 4.12 donne les résultats sur l'estimation du modèle à correction d'erreur. On constate que le coefficient associé à la force de rappel est négatif (-1.02) et significativement différent de zéro au seuil statistique de 5% (son t de Student est supérieur à 1.96 en valeur absolue). Il existe donc bien un mécanisme à correction d'erreur : à long terme les déséquilibres entre la consommation et le revenu se compensent de sorte que les deux séries ont une évolution similaire. On constate que le taux de croissance de la consommation dépend de façon positive du taux de croissance du revenu.

Remarque 4.2

Le package `arch` fournit une fonction `engle_granger` qui permet de tester la cointégration au sens de Engle et Granger pour 2 séries y_t et x_t .

```
# Test de cointégration au sens de Engle et Granger
from arch.unitroot import engle_granger
eg_test = engle_granger(eg_df.Y, eg_df.X, trend="c")
print(eg_test.summary())
```

```
##      Engle-Granger Cointegration Test
## =====
## Test Statistic                -5.399
## P-value                      0.000
## ADF Lag length                0
## Estimated Root   (+1)        -0.026
## -----
##
## Trend: Constant
## Critical Values: -3.25 (10%), -3.62 (5%), -4.38 (1%)
## Null Hypothesis: No Cointegration
## Alternative Hypothesis: Cointegration
## Distribution Order: 1
```


On visualise les résidus de cointégrations

```
# Représentation graphique
fig, axe = plt.subplots(figsize=(16,6))
eg_test.plot(axes=axe);
plt.show()
```

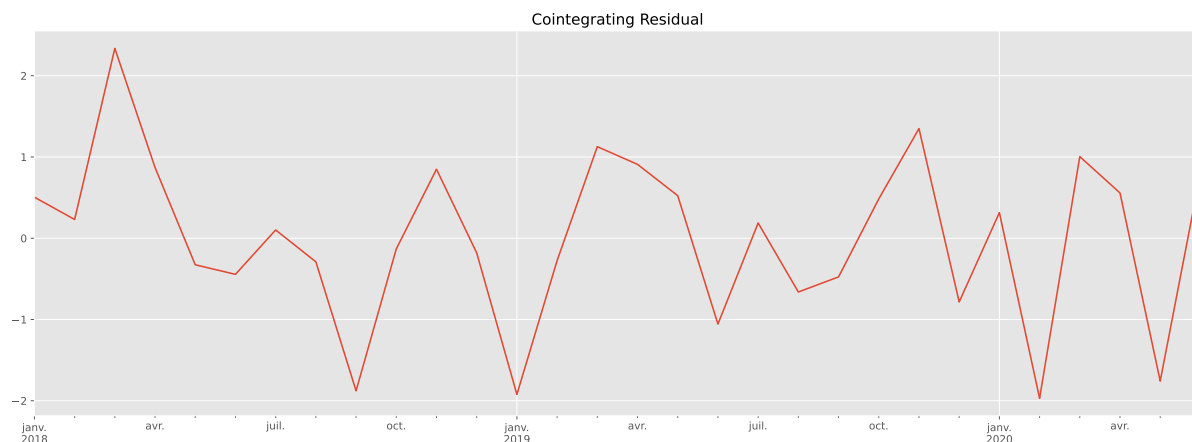


Figure 4.2 – Résidu de cointégration

on affiche le vecteur de cointégration

```
# Vecteur de cointégration
coint_vector = eg_test.cointegrating_vector
```

Table 4.13 – Vecteur de cointégration

Y	X	const
1	-0.5515969	-10.38444

On retrouve le même vecteur de cointégration à un signe près.

Exemple 4.2 Evolution du prix de l'essence

On considère les séries des prix au comptant du pétrole brut mesurés à Cushing (West Texas Intermediate et Brent). Ces données proviennent de [U.S. Energy Information Administration](#). Les séries ont été transformées en logarithme. Ces séries sont notées respectivement **Brent** et **WTI** et sont à fréquence mensuelle sur la période mai 1987 et janvier 2020.

La figure 4.3 indique que les deux séries semblent exhiber une tendance commune à la hausse. Ces deux séries paraissent non stationnaires et, du fait de leur apparente évolution similaire, il est légitime de s'intéresser à l'étude de la cointégration entre les deux variables.

```
# Chargement des données
from arch.data import crude
log_price = np.log(crude.load())
fig,axe = plt.subplots(figsize=(16,6))
axe.plot(log_price.WTI,color = "red",label = "WTI", linestyle = "--");
axe.plot(log_price.Brent,color = "black",label = "Brent");
```

```
axe.set_xlabel("mois");
axe.legend();
plt.show()
```

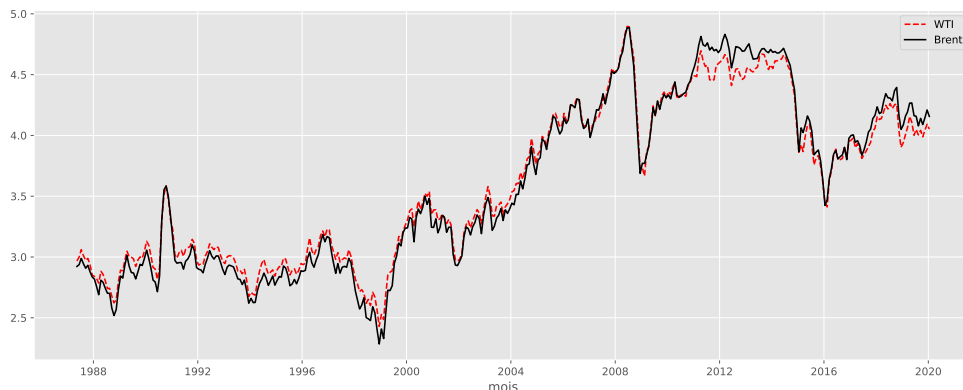


Figure 4.3 – Evolution des prix de l'essence

— **Etape 1** : Estimation de la relation de long terme

On estime la relation de long terme suivante : $WTI_t = \alpha_0 + \alpha_1 Brent_t + \varepsilon_t$

```
# Estimation de la relation de long terme
model2 = smf.ols('WTI~Brent', data = log_price).fit()
lm2_params = extractParams(model2, model_type = "lm")
```

Table 4.14 – Estimation de la relation de long terme entre WTI et Brent

	coef	std err	t	P> t	[0.025	0.975]
Intercept	0.3489	0.0136	25.6548	7.968e-86	0.3221	0.3756
Brent	0.9072	0.0037	244.3871	0.000e+00	0.8999	0.9145

Le tableau 4.14 donne les résultats de l'estimation de la relation de long terme entre WTI et Brent. Tous les coefficients sont significativement différents de zéro.

— **Etape 2** : Test DF sur les résidus de la relation de long terme

Nous vérifions que le résidu est bien stationnaire :

```
# Dickey - Fuller : Modèle [3]
resid2 = model2.resid
resid2_df3 = ADF(resid2,lags=0,trend = 'ct').regression
resid2_df3_params = extractParams(resid2_df3, model_type = "lm")
```

Table 4.15 – Coefficients du modèle [3] du test DF sur $\hat{\varepsilon}_t$

	coef	std err	t	P> t	[0.025	0.975]
Level.L1	-0.1505	0.0268	-5.6139	3.767e-08	-0.2032	-0.0978
const	0.0024	0.0028	0.8503	3.957e-01	-0.0031	0.0078
trend	0.0000	0.0000	-1.0056	3.152e-01	0.0000	0.0000

Table 4.16 – Valeurs critiques du modèle [3] pour $\hat{\varepsilon}_t$

	statistic	1pct	5pct	10pct
tau3	-5.6139	-3.98	-3.42	-3.13
phi2	10.5409	6.15	4.71	4.05
phi3	15.8093	8.34	6.30	5.36

La valeur empirique $t_{\hat{\rho}} = -5.6139 <$ aux trois valeurs critiques, on rejette l'hypothèse H_0 . Nous effectuons maintenant le test de l'hypothèse H_0^3 . La statistique $F_3 = 15.8093 >$ aux trois valeurs critiques pour Φ_3 . Nous rejetons les hypothèses H_0 et H_0^3 . Nous effectuons ensuite le test de nullité du paramètre b : la statistique de test $t_{\hat{b}} = -1.0056$ admet une p-value égale à 0.3152 supérieure à 5%, on accepte l'hypothèse de nullité de b . Enfin, nous effectuons le test nullité de la constante c : la statistique de test $t_{\hat{c}} = 0.8503$ admet une p-value égale à 0.3957 supérieure à 5%, on accepte l'hypothèse de nullité de c . En conclusion, le résidu est bien stationnaire, il existe donc un risque de cointégration entre les deux variables.

— **Etape 3** : Estimation du modèle à correction d'erreur

Modélisons le taux de croissance du prix WTI en fonction des résidus de la période précédente ($\hat{\varepsilon}_{t-1}$), du taux de croissance du prix Brent. Nous calculons, d'abord, le résidu décalé d'une période, soit : $\hat{\varepsilon}_{t-1} = WTI_{t-1} - 10.38 - 0.55Brent_{t-1}$, puis nous estimons par les MCO le modèle : $\Delta WTI_t = \phi \Delta Brent_t + \theta \hat{\varepsilon}_{t-1} + \eta_t$.

```
# Estimation du modèle à correction d'erreur
df2 = pd.concat([log_price.Brent.diff(), log_price.WTI.diff(), resid2.shift(1)],
                axis = 1)
df2.columns = ["Brent", "WTI", "ecm"]
ecm2 = smf.ols('WTI~Brent+ecm-1', data = df2.dropna()).fit()
ecm2_params = extractParams(ecm2, model_type = "lm")
```

Table 4.17 – Estimation du modèle à correction d'erreur (ECM)

	coef	std err	t	P> t	[0.025	0.975]
Brent	0.8794	0.0155	56.5908	3.853e-190	0.8489	0.9100
ecm	-0.1471	0.0265	-5.5540	5.175e-08	-0.1992	-0.0951

Le coefficient (force de rappel) de $\hat{\varepsilon}_{t-1}$ est bien significativement négatif, la représentation à correction d'erreur est validée.

Remarque 4.3

La méthode de Engle and Granger (1987) nous permet d'estimer aisément un modèle à correction d'erreur en deux étapes. Elle fournit également un certain nombre de tests de cointégration. L'inconvénient de cette approche est qu'elle ne permet pas de distinguer plusieurs vecteurs de cointégration. Ceci est problématique dès lors que l'on étudie simultanément d variables, avec $d > 2$. On sait en effet que si l'on analyse le comportement de d variables (avec $d > 2$), on peut avoir jusqu'à $(d-1)$ relations de cointégration ; l'approche de Engle and Granger (1987) ne nous permettant d'obtenir qu'une seule relation de cointégration. Afin de pallier cette difficulté, Søren Johansen (1988) a proposé une approche multivariée de la cointégration fondée sur la méthode du maximum de vraisemblance.

4.2.2 Approche multivariée de la cointégration : l'analyse de Johansen

La littérature économique concernant la théorie de l'équilibre démontre qu'une série macroéconomique stationnaire peut être le résultat d'une combinaison de variables non stationnaires, d'où l'importance actuelle de l'analyse de la cointégration à d variables. La généralisation de deux à d variables s'avère assez complexe du fait du nombre de possibilités de vecteurs de cointégration possibles. Nous faisons appel, dans ce cas, à la représentation vectorielle à correction d'erreur (*cf.* Bourbonnais, Terraza, et al. (2022)).

4.2.2.1 Le modèle vectoriel à correction d'erreur

Considérons le vecteur X_t de dimension d , dont les composantes sont toutes $I(1)$. La représentation VAR(p) de X_t est donnée par :

$$X_t = c + \Phi_1 X_{t-1} + \Phi_2 X_{t-2} + \dots + \Phi_p X_{t-p} + \varepsilon_t \quad (4.11)$$

où les matrices $\Phi_j (j = 1, \dots, p)$ sont de taille $(d \times d)$ et c un vecteur de dimension $(d \times 1)$

Examinons au préalable, le cas où $p = 2$, c'est - à - dire :

$$X_t = c + \Phi_1 X_{t-1} + \Phi_2 X_{t-2} + \varepsilon_t \quad (4.12)$$

Ce modèle peut s'écrire en différence première comme suit :

$$\Delta X_t = c + (\Phi_1 - I) X_{t-1} + \Phi_2 X_{t-2} + \varepsilon_t \quad (4.13)$$

Afin de faire apparaître des différences premières à droite de l'équation, nous ajoutons et soustrayons $\Phi_1 X_{t-2} - X_{t-2}$ de la manière suivante :

$$\begin{aligned} \Delta X_t &= c + (\Phi_1 - I) X_{t-1} + \Phi_2 X_{t-2} + (\Phi_1 X_{t-2} - X_{t-2} - \Phi_1 X_{t-2} + X_{t-2}) + \varepsilon_t \\ &= c + \Phi_1 X_{t-1} - X_{t-1} + \Phi_2 X_{t-2} + \Phi_1 X_{t-2} - X_{t-2} - \Phi_1 X_{t-2} + X_{t-2} + \varepsilon_t \\ \Delta X_t &= c + \Phi_1 (X_{t-1} - X_{t-2}) - (X_{t-1} - X_{t-2}) + (\Phi_1 + \Phi_2 - I) X_{t-2} + \varepsilon_t \end{aligned} \quad (4.14)$$

Or $\Delta X_{t-1} = X_{t-1} - X_{t-2}$.

En regroupant les termes en ΔX_{t-1} et après simplification, on obtient :

$$\Delta X_t = c + (\Phi_1 - I) \Delta X_{t-1} + (\Phi_1 + \Phi_2 - I) X_{t-2} + \varepsilon_t \quad (4.15)$$

Cependant, afin de faire figurer les relations de cointégration - qui relient entre elles les variables en niveau décalées d'une période - nous cherchons à écrire l'équation (4.14) en fonction de X_{t-1} , soit dans l'équation (4.13) nous pouvons ajouter et soustraire $\Phi_2 X_{t-1}$ et après simplification nous obtenons :

$$\Delta X_t = c + \Phi_2 \Delta X_{t-1} + (\Phi_1 + \Phi_2 - I) X_{t-1} + \varepsilon_t \quad (4.16)$$

En posant $\Theta_1 = -\Phi_2$ et $\pi = (\Phi_1 + \Phi_2 - I)$, l'équation (4.16) se réécrit sous la forme d'un modèle vectoriel à correction d'erreur ou VECM (*Vector Error Correction Model*) :

$$\Delta X_t = c + \Theta_1 \Delta X_{t-1} + \pi X_{t-1} + \varepsilon_t \quad (4.17)$$

Proposition 4.2 *Généralisation à p retards*

Le résultat précédent, equation (4.17) peut être généralisé dans le cas où l'on dispose d'une représentation VAR(p) à d variables. D'une part, elle peut s'écrire en différences premières :

$$\Delta X_t = c + (\Phi_1 - I) \Delta X_{t-1} + (\Phi_2 + \Phi_1 - I) \Delta X_{t-2} + \dots + (\Phi_{p-1} + \dots + \Phi_1 - I) \Delta X_{t-(p-1)} + \pi X_{t-p} + \varepsilon_t \quad (4.18)$$

ou encore en fonction de X_{t-1} :

$$\Delta X_t = c + \Theta_1 \Delta X_{t-1} + \Theta_2 \Delta X_{t-2} + \dots + \Theta_{p-1} \Delta X_{t-(p-1)} + \pi X_{t-p} + \varepsilon_t \quad (4.19)$$

Les matrices Θ_j sont fonctions des matrices Φ_j ($\Theta_j = -(\Phi_{j+1} + \dots + \Phi_p)$) et $\pi = \left(\sum_{j=1}^{j=p} \Phi_j - 1 \right)$.

Proposition 4.3 *Passage du VAR(p) au VECM(p-1)*

Considérons le processus X_t à d variables modélisé par un VAR(p) défini par l'équation (4.11). A cette équation, on soustrait X_{t-1} à droite et à gauche de l'équation :

$$\begin{aligned} X_t - X_{t-1} &= c + \Phi_1 X_{t-1} - X_{t-1} + \Phi_2 X_{t-2} + \dots + \Phi_p X_{t-p} + \varepsilon_t \\ \Delta X_t &= c + (-I + \Phi_1) X_{t-1} + \Phi_2 X_{t-2} + \dots + \Phi_p X_{t-p} + \varepsilon_t \end{aligned}$$

Ensuite ajoutons $\Phi_2 X_{t-1}$ et soustrayons $\Phi_2 X_{t-1}$ à droite de l'équation précédente :

$$\begin{aligned} \Delta X_t &= c + (-I + \Phi_1) X_{t-1} + \Phi_2 X_{t-1} - \Phi_2 X_{t-1} + \Phi_2 X_{t-2} + \dots + \Phi_p X_{t-p} + \varepsilon_t \\ \Delta X_t &= c + (-I + \Phi_1 + \Phi_2) X_{t-1} + (-\Phi_2) \Delta X_{t-1} + \Phi_3 X_{t-3} + \dots + \Phi_p X_{t-p} + \varepsilon_t \end{aligned}$$

En outre, ajoutons $\Phi_3 X_{t-1}$ et soustrayons $\Phi_3 X_{t-1}$ à droite de l'équation précédente :

$$\begin{aligned} \Delta X_t &= c + (-I + \Phi_1 + \Phi_2) X_{t-1} + (-\Phi_2) \Delta X_{t-1} + \Phi_3 X_{t-1} - \Phi_3 X_{t-1} + \Phi_3 X_{t-3} + \dots + \Phi_p X_{t-p} + \varepsilon_t \\ \Delta X_t &= c + (-I + \Phi_1 + \Phi_2 + \Phi_3) X_{t-1} + (-\Phi_2) \Delta X_{t-1} + (-\Phi_3) (X_{t-1} - X_{t-2} + X_{t-2} - X_{t-3}) + \\ &\quad \dots + \Phi_p X_{t-p} + \varepsilon_t \\ \Delta X_t &= c + (-I + \Phi_1 + \Phi_2 + \Phi_3) X_{t-1} + (-\Phi_2 - \Phi_3) \Delta X_{t-1} + (-\Phi_3) \Delta X_{t-2} + \dots + \Phi_p X_{t-p} + \varepsilon_t \end{aligned}$$

Et enfin nous ajoutons $\Phi_p X_{t-1}$ et soustrayons $\Phi_p X_{t-1}$ à droite de l'équation précédente :

$$\begin{aligned} \Delta X_t &= c + (-I + \Phi_1 + \Phi_2 + \Phi_3 + \dots + \Phi_{p-1}) X_{t-1} + (-\Phi_2 - \Phi_3 - \dots - \Phi_{p-1}) \Delta X_{t-1} + \\ &\quad (-\Phi_3 - \dots - \Phi_{p-1}) \Delta X_{t-2} + \dots + \Phi_p X_{t-p} + \varepsilon_t \\ \Delta X_t &= c + (-I + \Phi_1 + \Phi_2 + \Phi_3 + \dots + \Phi_{p-1}) X_{t-1} + (-\Phi_2 - \Phi_3 - \dots - \Phi_{p-1}) \Delta X_{t-1} + \\ &\quad (-\Phi_3 - \dots - \Phi_{p-1}) \Delta X_{t-2} + \dots + \Phi_p X_{t-1} - \Phi_p X_{t-1} + \Phi_p X_{t-p} + \varepsilon_t \\ \Delta X_t &= c + (-I + \Phi_1 + \Phi_2 + \Phi_3 + \dots + \Phi_{p-1} + \Phi_p) X_{t-1} + (-\Phi_2 - \Phi_3 - \dots - \Phi_{p-1}) \Delta X_{t-1} + \\ &\quad (-\Phi_3 - \dots - \Phi_{p-1}) \Delta X_{t-2} + \dots + (-\Phi_p) (X_{t-1} - X_{t-p}) + \varepsilon_t \\ \Delta X_t &= c + (-I + \Phi_1 + \Phi_2 + \Phi_3 + \dots + \Phi_{p-1} + \Phi_p) X_{t-1} + (-\Phi_2 - \Phi_3 - \dots - \Phi_{p-1}) \Delta X_{t-1} + \\ &\quad (-\Phi_3 - \dots - \Phi_{p-1}) \Delta X_{t-2} + \dots + \\ &\quad (-\Phi_p) (X_{t-1} - X_{t-2} + X_{t-2} - \dots - X_{t-(p-1)} + X_{t-(p-1)} - X_{t-p}) + \varepsilon_t \\ \Delta X_t &= c + (-I + \Phi_1 + \Phi_2 + \Phi_3 + \dots + \Phi_{p-1} + \Phi_p) X_{t-1} + (-\Phi_2 - \Phi_3 - \dots - \Phi_{p-1}) \Delta X_{t-1} + \\ &\quad (-\Phi_3 - \dots - \Phi_{p-1} - \Phi_p) \Delta X_{t-2} + \dots + (-\Phi_p) (X_{t-(p-1)} - X_{t-p}) + \varepsilon_t \\ &\quad (-\Phi_p) (X_{t-1} - X_{t-2} + X_{t-2} - \dots - X_{t-(p-1)} + X_{t-(p-1)} - X_{t-p}) + \varepsilon_t \\ \Delta X_t &= c + (-I + \Phi_1 + \Phi_2 + \Phi_3 + \dots + \Phi_{p-1} + \Phi_p) X_{t-1} + (-\Phi_2 - \Phi_3 - \dots - \Phi_{p-1}) \Delta X_{t-1} + \\ &\quad (-\Phi_3 - \dots - \Phi_{p-1} - \Phi_p) \Delta X_{t-2} + \dots + (-\Phi_p) \Delta X_{t-(p-1)} + \varepsilon_t \end{aligned}$$

En somme, l'écriture du modèle VECM(p-1) est :

$$\Delta X_t = c + \pi X_{t-1} + \Theta_1 \Delta X_{t-1} + \Theta_2 \Delta X_{t-2} + \dots + \Theta_{p-1} \Delta X_{t-(p-1)} + \varepsilon_t$$

avec :

$$\begin{cases} \pi = -(I - \Phi_1 - \Phi_2 - \dots - \Phi_p) = - \left(I - \sum_{j=1}^{j=p} \Phi_j \right) \\ \Theta_j = -(\Phi_{j+1} + \Phi_{j+2} + \dots + \Phi_p) = - \sum_{i=j+1}^{j=p} \Phi_i \end{cases}$$

πX_{t-1} est le premier décalage de combinaisons linéaires de variables en niveau non stationnaires ou de termes de correction d'erreur (ECT) qui représentent des relations à long terme entre des variables de niveau non stationnaires.

Exemple 4.3 *Représentation VECM(2) à trois variables*

Un modèle VECM(2) à trois variables x_t, y_t et z_t est représenté de la manière suivante :

$$\begin{bmatrix} \Delta x_t \\ \Delta y_t \\ \Delta z_t \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} + \begin{bmatrix} \pi_{11} & \pi_{12} & \pi_{13} \\ \pi_{21} & \pi_{22} & \pi_{23} \\ \pi_{31} & \pi_{32} & \pi_{33} \end{bmatrix} \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ z_{t-1} \end{bmatrix} + \begin{bmatrix} \theta_{11}^1 & \theta_{12}^1 & \theta_{13}^1 \\ \theta_{21}^1 & \theta_{22}^1 & \theta_{23}^1 \\ \theta_{31}^1 & \theta_{32}^1 & \theta_{33}^1 \end{bmatrix} \begin{bmatrix} \Delta x_{t-1} \\ \Delta y_{t-1} \\ \Delta z_{t-1} \end{bmatrix} + \begin{bmatrix} \theta_{11}^2 & \theta_{12}^2 & \theta_{13}^2 \\ \theta_{21}^2 & \theta_{22}^2 & \theta_{23}^2 \\ \theta_{31}^2 & \theta_{32}^2 & \theta_{33}^2 \end{bmatrix} \begin{bmatrix} \Delta x_{t-2} \\ \Delta y_{t-2} \\ \Delta z_{t-2} \end{bmatrix} + \begin{bmatrix} \varepsilon_{1,t} \\ \varepsilon_{2,t} \\ \varepsilon_{3,t} \end{bmatrix}$$

Remarque 4.4

Il est possible de retrouver la forme VAR(p) d'un processus à partir de sa forme vectorielle à correction d'erreur en appliquant les formules suivantes :

$$\begin{cases} \Phi_1 = \pi + I_d + \Theta_1 \\ \Phi_j = \Theta_j - \Theta_{j-1}, \quad j = 2, \dots, p-1 \\ \Phi_p = -\Theta_{p-1} \end{cases} \quad (4.20)$$

Tous les termes de l'équation (4.19) sont $I(0)$ sauf X_{t-p} qui est $I(1)$. On a donc un déséquilibre entre l'ordre d'intégration du membre de gauche et les termes du membre de droite. Pour que les deux membres de l'équation (4.19) soient $I(0)$, une condition nécessaire est que πX_{t-p} soit $I(0)$.

La matrice π peut s'écrire sous la forme $\pi = \alpha\beta'$ où le vecteur α (de dimension $d \times r$) est la force de rappel vers l'équilibre et β (de dimension $d \times r$) le vecteur dont les éléments sont les coefficients des relations de long terme des variables. En d'autres termes, β est la matrice de cointégration. Chaque combinaison linéaire représente donc une relation de cointégration. π est une matrice de rang r .

Proposition 4.4

- Si tous les éléments de π sont nuls (le rang de la matrice π est égal à 0 et donc $\sum_{j=1}^{j=p} \Phi_j = I$), alors nous ne pouvons pas retenir une spécification à correction d'erreur, nous estimons un VAR classique en différences premières afin d'éliminer les tendances.
- Si le rang de π est égal à d , cela implique alors que les variables sont toutes $I(0)$ et le problème de la cointégration ne se pose donc pas, il convient d'estimer un modèle VAR en niveau.
- Si le rang de la matrice π , noté r , est compris entre 1 et $d - 1$, alors il existe r relations de cointégration et la représentation ECM est valide, soit :

$$\Delta X_t = c + \sum_{j=1}^{j=p-1} \Theta_j \Delta X_{t-j} + \alpha ECT_{t-1} + \varepsilon_t \quad \text{avec} \quad ECT_t = \beta' X_t \quad (4.21)$$

Propriété 4.3 *Non unicité de la matrice π*

Il est important de noter que la décomposition de la matrice π de format $(d \times d)$ en produit de deux matrices de format $(d \times r)$, $\pi = \alpha\beta'$, n'est pas unique. En effet, pour toute matrice Q non singulière de format $(r \times r)$, on peut définir $\alpha^* = \alpha Q'$ et $\beta^* = \beta Q^{-1}$ et avoir $\pi = \alpha^* \beta^{*'}.$ Cette non unicité de la décomposition de π montre que les relations de cointégration ne sont pas uniques. Cependant, il est possible d'imposer des restrictions sur β et/ou α afin d'avoir des relations uniques.

Exemple 4.4 *Forme matricielle π en fonction de α et β*

L'identification de la matrice π dépend du nombre de vecteurs de cointégration de la manière suivante

- $r = 1$ (un vecteur de cointégration)

$$\begin{aligned} \pi X_{t-1} &= \begin{bmatrix} \pi_{11} & \pi_{12} & \pi_{13} \\ \pi_{21} & \pi_{22} & \pi_{23} \\ \pi_{31} & \pi_{32} & \pi_{33} \end{bmatrix} \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ z_{t-1} \end{bmatrix} = \begin{bmatrix} \alpha_{11} \\ \alpha_{21} \\ \alpha_{31} \end{bmatrix} \begin{bmatrix} 1 & -\beta_{11} & -\beta_{12} \end{bmatrix} \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ z_{t-1} \end{bmatrix} \\ &= \begin{bmatrix} \alpha_{11} \\ \alpha_{21} \\ \alpha_{31} \end{bmatrix} \begin{bmatrix} 1 \\ -\beta_{11} \\ -\beta_{12} \end{bmatrix} \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ z_{t-1} \end{bmatrix} = \alpha\beta' X_{t-1} \\ &= \begin{bmatrix} \alpha_{11} \\ \alpha_{21} \\ \alpha_{31} \end{bmatrix} [x_{t-1} - \beta_{11}y_{t-1} - \beta_{12}z_{t-1}] = \alpha ECT_{t-1} \end{aligned}$$

- $r = 2$ (deux vecteurs de cointégration)

$$\begin{aligned} \pi X_{t-1} &= \begin{bmatrix} \pi_{11} & \pi_{12} & \pi_{13} \\ \pi_{21} & \pi_{22} & \pi_{23} \\ \pi_{31} & \pi_{32} & \pi_{33} \end{bmatrix} \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ z_{t-1} \end{bmatrix} = \begin{bmatrix} \alpha_{11} & \alpha_{12} \\ \alpha_{21} & \alpha_{22} \\ \alpha_{31} & \alpha_{32} \end{bmatrix} \begin{bmatrix} 1 & -\beta_{11} & -\beta_{12} \\ 1 & -\beta_{21} & -\beta_{22} \end{bmatrix} \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ z_{t-1} \end{bmatrix} \\ &= \begin{bmatrix} \alpha_{11} & \alpha_{12} \\ \alpha_{21} & \alpha_{22} \\ \alpha_{31} & \alpha_{32} \end{bmatrix} \begin{bmatrix} 1 & 1 \\ -\beta_{11} & \beta_{21} \\ -\beta_{12} & \beta_{22} \end{bmatrix} \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ z_{t-1} \end{bmatrix} = \alpha\beta' X_{t-1} \\ &= \begin{bmatrix} \alpha_{11} \\ \alpha_{21} \\ \alpha_{31} \end{bmatrix} \begin{bmatrix} x_{t-1} - \beta_{11}y_{t-1} - \beta_{12}z_{t-1} \\ x_{t-1} - \beta_{21}y_{t-1} - \beta_{22}z_{t-1} \end{bmatrix} = \alpha ECT_{t-1} \end{aligned}$$

4.2.2.2 Estimateurs du maximum de vraisemblance

Selon les caractéristiques des données traitées, le processus VAR sous forme VECM peut connaître plusieurs variantes :

- Existence d'une constante, soit dans la relation de cointégration, soit dans le VAR ;
- Existence d'une tendance dans le VAR et/ou dans la relation de cointégration ;
- Prise en compte des variables exogènes de type indicatrice afin de corriger un mouvement saisonnier.

Ainsi, sur ce type de modèle, la méthode des MCO ne peut être appliquée car se posent des problèmes d'identification (le modèle n'est plus identifiable). Il convient alors d'employer la méthode du maximum de vraisemblance.

En tenant compte de la forme de $\pi = \alpha\beta'$, il est possible de réécrire l'équation (4.19) comme suit :

$$\Delta X_t = c + \Theta_1 \Delta X_{t-1} + \dots + \Theta_{p-1} \Delta X_{t-(p-1)} + \alpha\beta' X_{t-p} + \varepsilon_t \quad (4.22)$$

Pour estimer les différentes matrices, Søren Johansen (1988) propose d'utiliser la méthode du maximum de vraisemblance. Sous l'hypothèse de normalité de ε_t , la log-vraisemblance s'écrit (*cf.* Lardic and Mignon (2002)) :

$$\log L(\alpha, \beta, \Theta_1, \dots, \Theta_{p-1}, \Sigma_\varepsilon) = -\frac{dT}{2} \log(2\pi) - \frac{T}{2} \log[\det(\Sigma_\varepsilon)] - \frac{1}{2} \sum_{t=1}^{t=T} \varepsilon_t' \Sigma_\varepsilon^{-1} \varepsilon_t \quad (4.23)$$

où T est le nombre d'observations, d est le nombre de variables contenues dans X et $\det(\Sigma_\varepsilon)$ désigne le déterminant de la matrice de variance - covariance de ε_t .

— Concentration de la log - vraisemblance

On concentre la log - vraisemblance par rapport aux paramètres $\Theta_1, \Theta_2, \dots, \Theta_{p-1}$. Ceci revient à effectuer la régression de e_{0t} sur e_{pt} où e_{0t} et e_{pt} sont les résidus estimés des relations :

$$\Delta X_t = \Xi_{01} \Delta X_{t-1} + \dots + \Xi_{0(p-1)} \Delta X_{t-(p-1)} + e_{0t} \quad (4.24)$$

et

$$X_{t-p} = \Xi_{11} \Delta X_{t-1} + \dots + \Xi_{1(p-1)} \Delta X_{t-(p-1)} + e_{pt} \quad (4.25)$$

avec $\Xi_{0i} = (\Theta_i - \alpha\beta' \Theta_i)$ et $\Xi_{1i} = \Theta_i$ pour $i = 1, \dots, p-1$.

Ces deux équations découlent de la décomposition de l'équation (4.19) de la manière suivante :

$$\begin{aligned} \Delta X_t - \alpha\beta' X_{t-p} &= \Theta_1 \Delta X_{t-1} + \dots + \Theta_{p-1} \Delta X_{t-(p-1)} + \varepsilon_t \\ &= \Theta_1 \Delta X_{t-1} + \dots + \Theta_{p-1} \Delta X_{t-(p-1)} + \alpha\beta' [\Theta_1 \Delta X_{t-1} + \dots + \Theta_{p-1} \Delta X_{t-(p-1)}] - \\ &\quad \alpha\beta' [\Theta_1 \Delta X_{t-1} + \dots + \Theta_{p-1} \Delta X_{t-(p-1)}] + \varepsilon_t \\ &= [\Theta_1 - \alpha\beta' \Theta_1] \Delta X_{t-1} + \dots + [\Theta_{p-1} - \alpha\beta' \Theta_{p-1}] \Delta X_{t-(p-1)} + \\ &\quad \alpha\beta' [\Theta_1 \Delta X_{t-1} + \dots + \Theta_{p-1} \Delta X_{t-(p-1)}] + \varepsilon_t \end{aligned}$$

On extrait la valeur de ε_t en fonction des autres variables :

$$\begin{aligned}
\varepsilon_t &= e_{0t} - \alpha\beta' e_{pt} \\
&= \Delta X_t - [\Theta_1 - \alpha\beta' \Theta_1] \Delta X_{t-1} + \dots + [\Theta_{p-1} - \alpha\beta' \Theta_{p-1}] \Delta X_{t-(p-1)} - \\
&\quad \alpha\beta' [X_{t-p} - \Theta_1 \Delta X_{t-1} + \dots + \Theta_{p-1} \Delta X_{t-(p-1)}]
\end{aligned}$$

Les régressions précédentes nous permettent d'obtenir des estimateurs des matrices Θ_i pour $i = 1, \dots, p-1$. On peut alors réécrire la log - vraisemblance comme suit :

$$\begin{aligned}
\log L(\alpha, \beta, \Sigma_\varepsilon) &= -\frac{dT}{2} \log(2\pi) - \frac{T}{2} \log[\det(\Sigma_\varepsilon)] \\
&\quad - \frac{1}{2} \sum_{t=1}^{t=T} (e_{0t} - \alpha\beta' e_{pt})' \Omega^{-1} (e_{0t} - \alpha\beta' e_{pt})
\end{aligned} \tag{4.26}$$

On maximise cette expression par rapport à α et Σ_ε pour β fixé. Dans la mesure où l'on a un modèle de régressions empilées avec les mêmes variables explicatives ($\beta' e_{pt}$), les estimateurs des moindres carrés généralisés coïncident avec ceux des moindres carrés ordinaires. On en déduit donc :

$$\hat{\alpha}(\beta) = -S_{0p}\beta (\beta' S_{pp}\beta)' \tag{4.27}$$

et

$$\hat{\Sigma}_\varepsilon(\beta) = S_{00} - S_{0p}\beta (\beta' S_{pp}\beta)^{-1} \beta' S_{p0} \tag{4.28}$$

où $S_{kl} = \frac{1}{T} \sum_{t=1}^{t=T} e_{kt} e_{lt}'$ pour $k, l = 0, \dots, p$. L'expression de S_{kl} nous donne les matrices de variance - covariance de e_{0t} et e_{pt} .

Søren Johansen (1988) a montré que la concentration de la log - vraisemblance par rapport à Ω permet d'obtenir l'expression suivante² :

$$\log L(\beta) = -\frac{dT}{2} \log(2\pi) - \frac{T}{2} \log[\det(\hat{\Sigma}_\varepsilon(\beta))] - \frac{dT}{2} \tag{4.29}$$

La log - vraisemblance ne dépend plus que de $\det(\hat{\Sigma}_\varepsilon(\beta))$.

— Estimation des vecteurs de cointégration

Maximiser la log - vraisemblance par rapport $\hat{\Sigma}_\varepsilon(\beta)$ est équivalent à résoudre le programme de minimisation suivant :

$$\min_{\beta} \det(\hat{\Sigma}_\varepsilon(\beta)) \Leftrightarrow \min_{\beta} \det[S_{00} - S_{0p}\beta(\beta' S_{pp}\beta)^{-1} \beta' S_{p0}] \tag{4.30}$$

La solution de ce problème est donnée par Anderson (1984) et se déduit d'une recherche de valeurs propres et de vecteurs propres. Ainsi, pour obtenir les vecteurs de cointégration recherchés, il faut trouver les d valeurs propres de l'équation :

$$\det[\lambda S_{pp} - S_{p0} S_{00}^{-1} S_{0p}] = 0 \tag{4.31}$$

2. Johansen (1988) a en effet montré que $\frac{1}{2} \sum_{t=1}^{t=T} (e_{0t} - \alpha\beta' e_{pt})' \Sigma_\varepsilon^{-1} (e_{0t} - \alpha\beta' e_{pt}) = \frac{dT}{2}$

La résolution de l'équation (4.31) nous donne les d valeurs propres $(\hat{\lambda}_1, \dots, \hat{\lambda}_d)$ avec $\hat{\lambda}_1 > \dots > \hat{\lambda}_d$ et les d vecteurs propres associés $(\hat{u}_1, \dots, \hat{u}_d)$ avec la condition de normalisation $\widehat{U}' S_{pp} \widehat{U} = I$ où \widehat{U} est la matrice des vecteurs propres estimés.

On peut alors montrer que les estimateurs du maximum de vraisemblance des vecteurs de cointégration sont les vecteurs propres $(\hat{u}_1, \dots, \hat{u}_r)$ associés aux r plus grandes valeurs propres solutions de (4.31). Plus précisément, l'estimateur de α est donnée par :

$$\hat{\alpha} = -S_{0p} \hat{\beta} (\hat{\beta}' S_{pp} \hat{\beta})^{-1} = -S_{0p} \hat{\beta} \quad (4.32)$$

et

$$\widehat{\Omega} = S_{00} - S_{0p} \hat{\beta} \hat{\beta}' S_{p0} = S_{00} - \hat{\alpha} \hat{\alpha}' \quad (4.33)$$

et la log - vraisemblance maximale s'écrit :

$$\log L_{\max} = -\frac{dT}{2} \log(2\pi) - \frac{T}{2} \log[\det(S_{00})] - \frac{T}{2} \sum_{i=1}^{i=r} \log(1 - \hat{\lambda}_i) - \frac{dT}{2} \quad (4.34)$$

Les r vecteurs de cointégration sont donnés par les r vecteurs propres les plus significatifs.

4.2.2.3 Détermination du nombre de vecteurs de cointégration

Pour déterminer le nombre de relations de cointégration, Søren Johansen (1988) propose deux tests statistiques : le test de la trace et le test de la valeur propre :

1) Test de la trace

Il s'agit d'un test du rapport de maximum de vraisemblance consistant à calculer la statistique suivante :

$$TR(q) = -T \sum_{i=q+1}^d (1 - \hat{\lambda}_i) \quad (4.35)$$

Les hypothèses du test sont les suivantes :

$$\begin{cases} H_0 : \text{vecteurs de cointégration} \leq q \\ H_1 : \text{vecteurs de cointégration} \geq q + 1 \end{cases}$$

L'hypothèse nulle testée est $r \leq q$, c'est - à - dire qu'il existe au plus r vecteurs de cointégration. Ce test revient à tester le rang de la matrice π puisque tester l'existence de r vecteurs de cointégration revient à tester l'hypothèse nulle : $\text{rang}(\pi) = r$. On rejette l'hypothèse nulle de r relations de cointégration lorsque la statistique TR est supérieure à sa valeur critique. Les valeurs critiques de la statistique TR ont été tabulées par Søren Johansen and Juselius (1990) puis Osterwald-Lenum (1992).

Ce test de Johansen fonctionne par exclusion d'hypothèses alternatives :

- rang de la matrice π égal 0 ($r = 0$), soit $H_0 : r = 0$ contre $H_1 : r > 0$; si H_0 est refusée, on passe au test suivant³(si $TR >$ à la valeur critique lue dans la table, on rejette H_0) ;

3. Dans le cas contraire, la procédure est arrêtée, et le rang de la matrice est $r = 0$.

- rang de la matrice π égal 1 ($r = 1$), soit $H_0 : r = 1$ contre $H_1 : r > 1$; si H_0 est refusée, on passe au test suivant ;
- rang de la matrice π égal 2 ($r = 2$), soit $H_0 : r = 2$ contre $H_1 : r > 2$; si H_0 est refusée, on passe au test suivant, etc.

Si, après avoir refusé les différentes hypothèses H_0 à la fin de la procédure, on teste $H_0 : r = d-1$ contre $H_1 : r = d$ et que l'on soit amené à refuser H_0 , alors le rang de la matrice est $r = d$ et il n'existe pas de relation de cointégration car les variables sont toutes $I(0)$.

Remarque 4.5

Le test de la trace permet de déterminer le nombre de relations de cointégration, mais n'indique pas les variables qui sont cointégrées.

2) Test de la valeur propre maximale

La statistique de test est donnée par :

$$VP_{\max}(q) = -T \log(1 - \hat{\lambda}_{q+1}) \quad (4.36)$$

Les hypothèses du test sont les suivantes :

$$\begin{cases} H_0 : \text{Il existe } q \text{ vecteurs de cointégration} \\ H_1 : \text{Il existe } q + 1 \text{ vecteurs de cointégration} \end{cases}$$

On test l'hypothèse nulle $r = q$ contre l'hypothèse alternative $r = q + 1$. Le test de la trace est le plus utilisé que celui de la valeur propre maximale.

Remarque 4.6

Les valeurs critiques pour les tests changent dès lors que l'on introduit une constante et/ou une tendance. Quatre spécifications sont possibles :

- Cas 1 : Absence de constante dans le modèle à correction d'erreur (les séries n'ont pas de tendance linéaire) et dans les relations de cointégration.
- Cas 2 : Constante dans le modèle à correction d'erreur (les séries sont caractérisées par des tendances linéaires) et dans les relations de cointégration.
- Cas 3 : Constante dans les relations de cointégration mais pas dans le modèle à correction d'erreur (les séries n'ont pas de tendance linéaire).
- Cas 4 : Constante dans le modèle à correction d'erreur (les séries sont caractérisées par une tendance linéaire) et constante et tendance dans les relations de cointégration.
- Cas 5 : Trendance quadratique dans le modèle à correction d'erreur et tendances linéaires dans les relations de cointégration.

Le choix d'une de ces spécifications s'effectue en fonction des données et de la forme supposée de la tendance. Les modèles testés dans les différentes spécifications s'écrivent alors :

- Cas 1 : $\pi X_{t-1} = \alpha \beta' X_{t-1}$
- Cas 2 : $\pi X_{t-1} = \alpha(\beta' X_{t-1} + \mu)$
- Cas 3 : $\pi X_{t-1} = \alpha(\beta' X_{t-1} + \mu) + \lambda$
- Cas 4 : $\pi X_{t-1} = \alpha(\beta' X_{t-1} + \mu + \alpha t) + \lambda$
- Cas 5 : $\pi X_{t-1} = \alpha(\beta' X_{t-1} + \mu + \alpha t) + \lambda + \beta t$

En cas de divergence entre le test de la trace et celui de la valeur propre maximale, nous privilégions le test de la trace dont la puissance est la plus élevée.

Sous Python, la fonction `coint_johansen` de la librairie Statsmodels implémente le test de cointégration au sens de Johansen.

Exemple 4.5 Tests de cointégration et estimation d'un modèle VECM

On considère les séries de taux d'intérêt à court terme des villages NGUI, NDAH et FIALAH dans le groupement FONGO - NDENG. Ces données sont à fréquence mensuelle sur la période janvier 2018 à juin 2020. On demande de tester une éventuelle cointégration et d'estimer un modèle VAR ou un modèle vectoriel à correction d'erreur s'il y a lieu.

```
# Représentation graphique
fig, axe = plt.subplots(figsize=(16,6))
joh_df.plot(ax=axe);
plt.show()
```

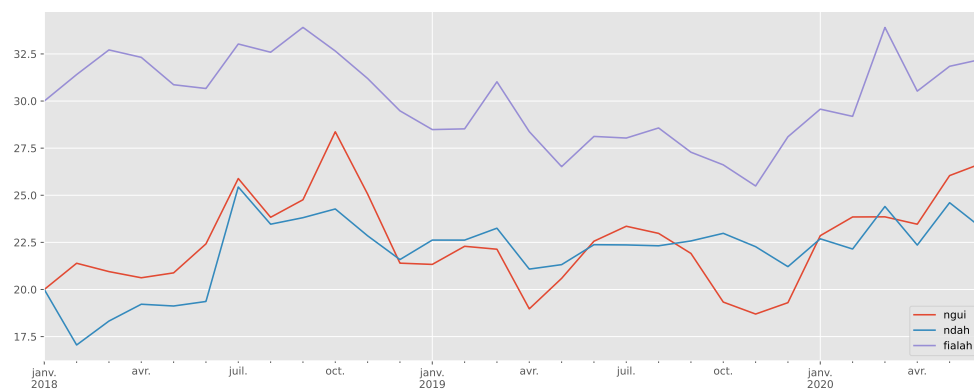


Figure 4.4 – Evolution des taux d'intérêts

On constate graphiquement (Figure 4.4) que ces séries sont caractérisées par une tendance générale à la hausse. Par ailleurs, des tests de racine uniatrice ont été effectués et ont montré que les trois séries sont intégrées d'ordre 1.

— **Etape 1** : Détermination du nombre de retards du VAR en niveau

Compte tenu du faible nombre d'observations, nous effectuons le calcul des critères d'information pour un retard maximal fixé à 3.

```
# Choix de p du VAR en niveau
from statsmodels.tsa.api import VAR
pchoice = VAR(joh_df).select_order(3).summary()
print(pchoice)

## VAR Order Selection (* highlights the minimums)
## =====
##      AIC      BIC      FPE      HQIC
## -----
## 0      3.264      3.408      26.16      3.307
```

```
## 1      2.232      2.808*      9.381      2.403
## 2      2.099*      3.107      8.458*      2.398*
## 3      2.264      3.704      10.75      2.692
## -----
```

Le retard retenu est donc de 2 (minimum des critères AIC, FPE, HQ), nous procédons donc au test de Johansen sur un VECM(1).

On peut réaliser la même opération en utilisant la fonction `select_order` du module `vecm` contenu dans la librairie `Statsmodels`.

```
# Choix de p-1 du VECM
from statsmodels.tsa.vector_ar.vecm import select_order
lag_order = select_order(data=joh_df, maxlags=2)
print(lag_order.summary())
```

```
## VECM Order Selection (* highlights the minimums)
## =====
##      AIC      BIC      FPE      HQIC
## -----
## 0      2.232      2.808*      9.381      2.403
## 1      2.099*      3.107      8.458*      2.398*
## 2      2.264      3.704      10.75      2.692
## -----
```

La ligne $p = 1$ du VECM est identique à celle $p = 2$ du VAR en niveau.

— **Etape 2** : Test de cointégration au sens de Johansen

Nous procédons au test de cointégration au sens de Johansen sous deux hypothèses :

a) Existence d'une constante dans la relation de long terme et non dans les données

Premier test : Rang de la matrice π égal 0 ($r = 0$), soit $H_0 : r = 0$ contre $H_1 : r > 0$.

```
# Test de Johansen : spécification b
from statsmodels.tsa.vector_ar.vecm import coint_johansen
joh_test = coint_johansen(endog=joh_df, det_order=0, k_ar_diff=1)
```

Les valeurs propres de la matrice π estimée par le maximum de vraisemblance sont consignées dans le tableau 4.18

```
# Valeurs propres
eig_df = pd.DataFrame({
    "eig1" : joh_test.eig[0], "eig2":joh_test.eig[1], "eig3":joh_test.eig[2]
}, index = ["eigen value"])
```

Table 4.18 – Valeurs propres $\hat{\lambda}$

	$\hat{\lambda}_1$	$\hat{\lambda}_2$	$\hat{\lambda}_3$
eigen value	0.604	0.229	0.139

Calculons la statistique de Johansen : $TR = -T \sum_{i=q+1}^{i=d} \log(1 - \hat{\lambda}_i)$ pour $r = 0$.

```
# Statistique de Johansen : Test de la trace pour r= 0
tr0 = -(len(joh_df)-2)*np.sum([np.log(1-x) for x in joh_test.eig])
print("statistique de trace (r = 0) : %.2f"%(tr0))
```

```
## statistique de trace (r = 0) : 37.39
```

Les valeurs critiques pour $r = 0$ sont consignées dans le tableau 4.19

```
# Les valeurs critiques pour r = 0
ctr0 = pd.DataFrame({
    "90%":joh_test.cvt[0][0], "95%":joh_test.cvt[0][1], "99%":joh_test.cvt[0][2]
},index = ["critical value"])
```

Table 4.19 – Valeurs critiques pour $r=0$

	90%	95%	99%
critical value	27.067	29.796	35.463

Les valeurs critiques sont égales à 29.796 pour un seuil de 5% et 35.463 pour un seuil de 1%; on rejette donc l'hypothèse H_0 , le rang de la matrice n'est pas 0.

Second test : Rang de la matrice π égal 1 ($r = 1$), soit $H_0 : r = 1$ contre $H_1 : r > 1$

```
# Statistique de Johansen : Test de la trace
tr1 = -(len(joh_df)-2)*np.sum([np.log(1-x) for x in joh_test.eig[1:]])
print("statistique de trace (r = 1) : %.2f"%(tr1))
```

```
## statistique de trace (r = 1) : 11.47
```

Les valeurs critiques pour $r = 1$ sont consignées dans le tableau 4.20

```
# Les valeurs critiques pour r = 1
ctr1 = pd.DataFrame({
    "90%":joh_test.cvt[1][0], "95%":joh_test.cvt[1][1], "99%":joh_test.cvt[1][2]
},index = ["critical value"])
```

Table 4.20 – Valeurs critiques pour $r = 1$

	90%	95%	99%
critical value	13.429	15.494	19.935

Les valeurs critiques sont égales à 15.494 pour un seuil de 5% et 19.935 pour un seuil de 1%; on ne peut rejeter l'hypothèse H_0 ni à 5% ni à 1%, on considère donc que le rang de la matrice π est égal à 1.

Remarque 4.7

Les attributs `res.eig`, `res.lr1` (respectivement `res.lr2`) et `res.cvt` (respectivement `res.cvm`) retournent respectivement les valeurs propres de la matrice π , la statistique de Johansen associé au test de la trace (respectivement du test de la valeur propre maximale) et les valeurs critiques aux niveaux de 90%, 95% et 99% associées au test de la trace (respectivement au test de la valeur propre maximale). Le tableau 4.21 donne le résultat global sur le test de la trace.

```
#Extraction des résultats du test de Johansen
def joh_output(res):
    # Test de la trace
    trace = pd.DataFrame(res.cvt); trace.insert(0, "TR", list(res.lr1))
    trace.insert(0, "eigen_value", list(res.eig))
    trace.insert(0, "rank", ["r = "+str(x) for x in list(res.ind)])
    trace.rename(columns = {0:'90%', 1: '95%', 2:'99%'}, inplace=True)
    # Test de la valeur propre maximale
    maxeig = pd.DataFrame(res.cvm); maxeig.insert(0, "TR", list(res.lr2))
    maxeig.insert(0, "eigen_value", list(res.eig))
    maxeig.insert(0, "rank", ["r = "+str(x) for x in list(res.ind)])
    maxeig.rename(columns = {0:'90%', 1: '95%', 2:'99%'}, inplace=True)
    return maxeig, trace
# Application
maxeig_test, trace_test = joh_output(joh_test)
```

Table 4.21 – Test de la trace (Spécification b))

rank	Eigen value	TR	90%	95%	99%
r = 0	0.6037495	37.38888	27.0669	29.7961	35.4628
r = 1	0.2293111	11.46904	13.4294	15.4943	19.9349
r = 2	0.1385500	4.17587	2.7055	3.8415	6.6349

Nous acceptons donc l'hypothèse d'une relation de cointégration.

Une autre façon de déterminer le nombre de relation de cointégration sous Python est d'utiliser la fonction `select_coint_rank` du module `vecm` de la librairie `Statsmodels`. Elle est plus pratique et plus explicite que la précédente.

```
# Détermination du nombre de relation de cointégration
from statsmodels.tsa.vector_ar.vecm import select_coint_rank
rank = select_coint_rank(joh_df,det_order = 0,k_ar_diff = 1,method = 'trace',
                        signif=0.05)
print(rank.summary())

## Johansen cointegration test using trace test statistic with 5% significance level
## =====
## r_0 r_1 test statistic critical value
## -----
##    0    3          37.39          29.80
##    1    3          11.47          15.49
## -----
```

On peut remplacer l'option `trace` par `maxeig` afin d'obtenir le test de la valeur propre maximale.

b) Existence d'une constante dans la relation de long terme et dans les données

Les résultats du test de trace sont consignés dans le tableau 4.22 :

```
# Test de Johansen : spécification c
joh2_test = coint_johansen(endog=joh_df, det_order=1, k_ar_diff=1)
maxeig2_test, trace2_test = joh_output(joh2_test)
```

Table 4.22 – Test de la trace (Spécification c))

rank	Eigenvalue	TR	90%	95%	99%
r = 0	0.5987968	36.615344	32.0645	35.0116	41.0815
r = 1	0.2247222	11.043300	16.1619	18.3985	23.1485
r = 2	0.1305285	3.916353	2.7055	3.8415	6.6349

Là encore, on constate que le rang de la matrice π n'est pas 0, mais qu'en revanche, on ne peut rejeter l'hypothèse H_0 ni à 5% ni à 1% dans l'hypothèse d'un rang de la matrice π égal à 1.

Le test de valeur propre maximale corrobore les résultats précédents (*cf.* Tableau 4.23) :

```
# Test de la valeur propre maximale
maxeig2_test = (maxeig2_test
.replace({'rank': {"r = 0": "None", "r = 1": "At most 1", "r = 2": "At most 2"}}))
```

Table 4.23 – Test de la valeur propre maximale (Spécification c))

No. of CE(s)	Eigenvalue	VP	90%	95%	99%
None	0.5987968	25.572044	21.8731	24.2522	29.2631
At most 1	0.2247222	7.126947	15.0006	17.1481	21.7465
At most 2	0.1305285	3.916353	2.7055	3.8415	6.6349

L'hypothèse d'une seule relation de cointégration est validée.

— **Etape 3** : Estimation du modèle VECM

Les deux spécifications précédentes sont estimées avec ou sans constante dans les données et donc, dans les deux cas, avec une seule relation de cointégration entre $ngui_t$, $ndah_t$ et $fialah_t$.

a) Estimation de la première spécification (specification b)) :

```
# Estimation du VECM(1) : constant outside the cointegration relation (co)
from statsmodels.tsa.vector_ar.vecm import VECM
vecm1 = VECM(joh_df, k_ar_diff=1, coint_rank=1, deterministic= "co").fit()
print(vecm1.summary())

## Det. terms outside the coint. relation & lagged endog. parameters for equation ngui
## =====
##          coef      std err          z      P>|z|      [0.025      0.975]
## -----
## const          -10.9343        3.766       -2.903      0.004      -18.316       -3.553
## L1.ngui           0.7661        0.247        3.096      0.002         0.281        1.251
## L1.ndah          -0.7290        0.250       -2.916      0.004       -1.219       -0.239
## L1.fialah        -0.2607        0.219       -1.189      0.235       -0.691        0.169
## Det. terms outside the coint. relation & lagged endog. parameters for equation ndah
## =====
##          coef      std err          z      P>|z|      [0.025      0.975]
## -----
## const           4.1740        3.240        1.288      0.198       -2.177       10.525
## L1.ngui           0.1010        0.213        0.474      0.635       -0.316        0.518
## L1.ndah          -0.3559        0.215       -1.655      0.098       -0.777        0.066
## L1.fialah        -0.0818        0.189       -0.433      0.665       -0.452        0.288
```



```
## Det. terms outside the coint. relation & lagged endog. parameters for equation fialah
## =====
##          coef      std err          z      P>|z|      [0.025      0.975]
## -----
## const          0.1196      3.445      0.035      0.972      -6.632      6.871
## L1.ngui          0.5293      0.226      2.339      0.019      0.086      0.973
## L1.ndah         -0.4579      0.229      -2.003      0.045      -0.906      -0.010
## L1.fialah        -0.2458      0.201      -1.225      0.221      -0.639      0.147
##          Loading coefficients (alpha) for equation ngui
## =====
##          coef      std err          z      P>|z|      [0.025      0.975]
## -----
## ec1           -0.8700      0.293      -2.965      0.003      -1.445      -0.295
##          Loading coefficients (alpha) for equation ndah
## =====
##          coef      std err          z      P>|z|      [0.025      0.975]
## -----
## ec1           0.3066      0.252      1.215      0.224      -0.188      0.801
##          Loading coefficients (alpha) for equation fialah
## =====
##          coef      std err          z      P>|z|      [0.025      0.975]
## -----
## ec1           0.0089      0.268      0.033      0.974      -0.517      0.535
##          Cointegration relations for loading-coefficients-column 1
## =====
##          coef      std err          z      P>|z|      [0.025      0.975]
## -----
## beta.1          1.0000          0          0          0.000          1.000          1.000
## beta.2         -0.8619      0.093      -9.305      0.000      -1.043      -0.680
## beta.3         -0.5395      0.075      -7.216      0.000      -0.686      -0.393
## =====
```

Cette première spécification est rejetée du fait de la non significativité des constantes dans les équations associées à `ndah` et `fialah`.

b) Estimation de la deuxième spécification (spécification c))

```
# Estimation du VECM(1) : constant within the cointegration relation (ci)
from statsmodels.tsa.vector_ar.vecm import VECM
vecm2 = VECM(joh_df, k_ar_diff=1, coint_rank=1, deterministic="ci").fit()
print(vecm2.summary())

## Det. terms outside the coint. relation & lagged endog. parameters for equation ngui
## =====
##          coef      std err          z      P>|z|      [0.025      0.975]
## -----
## L1.ngui          0.7655      0.251      3.052      0.002      0.274      1.257
## L1.ndah         -0.7129      0.253      -2.817      0.005      -1.209      -0.217
## L1.fialah        -0.2582      0.222      -1.161      0.245      -0.694      0.178
## Det. terms outside the coint. relation & lagged endog. parameters for equation ndah
## =====
```

```

##              coef      std err          z      P>|z|      [0.025      0.975]
## -----
## L1.ngui         0.1013        0.216        0.470        0.638        -0.321        0.524
## L1.ndah        -0.3400        0.217       -1.564        0.118        -0.766        0.086
## L1.fialah       -0.0811        0.191       -0.425        0.671        -0.456        0.293
## Det. terms outside the coint. relation & lagged endog. parameters for equation fialah
## =====
##              coef      std err          z      P>|z|      [0.025      0.975]
## -----
## L1.ngui         0.5295        0.226        2.341        0.019         0.086        0.973
## L1.ndah        -0.4577        0.228       -2.006        0.045        -0.905       -0.010
## L1.fialah       -0.2459        0.200       -1.226        0.220        -0.639        0.147
##              Loading coefficients (alpha) for equation ngui
## =====
##              coef      std err          z      P>|z|      [0.025      0.975]
## -----
## ec1            -0.8466        0.296       -2.858        0.004        -1.427       -0.266
##              Loading coefficients (alpha) for equation ndah
## =====
##              coef      std err          z      P>|z|      [0.025      0.975]
## -----
## ec1            0.3248        0.255        1.276        0.202        -0.174        0.824
##              Loading coefficients (alpha) for equation fialah
## =====
##              coef      std err          z      P>|z|      [0.025      0.975]
## -----
## ec1            0.0090        0.267         0.034        0.973        -0.515        0.532
##              Cointegration relations for loading-coefficients-column 1
## =====
##              coef      std err          z      P>|z|      [0.025      0.975]
## -----
## beta.1          1.0000          0          0          0.000         1.000         1.000
## beta.2         -0.8660         0.093       -9.326        0.000        -1.048       -0.684
## beta.3         -0.5383         0.075       -7.182        0.000        -0.685       -0.391
## const          12.9187         3.071         4.207        0.000         6.900        18.938
## =====

```

— Etape 4 : Validation de la représentation VECM

On remarque que les coefficients des termes de rappel ont bien le signe attendu (cependant, seul le terme de rappel de l'équation associé à Δngui est significatif) et les coefficients de la relation de long terme sont significatifs.

Le fait que seul le coefficient α_1 soit significativement différent de 0, indique que les variables `ndah` et `fialah` sont faiblement exogènes, ce qui doit être testé avec un test du ratio de vraisemblance. Cependant, Python ne dispose d'aucune fonction susceptible de réaliser ce test pour un modèle VECM. Notre analyse s'arrête là.

Représentation alternative

— Coefficients du VAR(2) en fonction de ceux du VECM(1)

Transformons notre VECM(1) en un VAR(2) en niveau. On applique les formules suivantes pour

Φ_1 et Φ_2 :

$$\begin{cases} \Phi_1 = \alpha\beta' + I_3 + \Theta_1 \\ \Phi_2 = -\Theta_1 \end{cases}$$

```
# Coefficients du VAR(2)
alpha, beta, theta = vecm2.alpha, vecm2.beta, vecm2.gamma
pi = alpha.dot(beta.T); phi1 = pi + np.identity(3)+theta; phi2 = - theta
var_params = pd.DataFrame(
    np.append(phi1,phi2,axis=1),index = ["ngui.L0","ndah.L0", "fialah.L0"],
    columns = ["ngui.L1", "ndah.L1","fialah.L1","ngui.L2", "ndah.L2", "fialah.L2"])
```

Table 4.24 – Coefficients associés au VAR(2) obtenus par calcul manuel

	ngui.L1	ndah.L1	fialah.L1	ngui.L2	ndah.L2	fialah.L2
ngui.L0	0.919	0.020	0.198	-0.765	0.713	0.258
ndah.L0	0.426	0.379	-0.256	-0.101	0.340	0.081
fialah.L0	0.538	-0.465	0.749	-0.530	0.458	0.246

Une façon facile et rapide d'obtenir les coefficients associés à la représentation VAR(p) d'un VECM(p-1) est d'utiliser la méthode `res.var_rep` du modèle vectoriel à correction d'erreur entraîné.

```
# Coefficient du modèle VAR(2) associé au VECM(1)
var2_params = pd.DataFrame(
    np.append(np.array(vecm2.var_rep[0]),np.array(vecm2.var_rep[1]),axis=1),
    columns = ["ngui.L1", "ndah.L1","fialah.L1","ngui.L2", "ndah.L2", "fialah.L2"],
    index = ["ngui.L0","ndah.L0", "fialah.L0"])
```

Table 4.25 – Coefficients associés au VAR(2) obtenus automatiquement

	ngui.L1	ndah.L1	fialah.L1	ngui.L2	ndah.L2	fialah.L2
ngui.L0	0.919	0.020	0.198	-0.765	0.713	0.258
ndah.L0	0.426	0.379	-0.256	-0.101	0.340	0.081
fialah.L0	0.538	-0.465	0.749	-0.530	0.458	0.246

— Forme $VMA(\infty)$ associée à la représentation VECM

Nous devons noter que les processus VAR(p) instables, intégrés ou cointégrés n'ont pas une représentation $VMA(\infty)$ valide. Par conséquent, les valeurs des coefficients fournies par la méthode `res.ma_rep` doivent être analysées avec prudence.

Exemple 4.6 Représentation $VMA(\infty)$ du modèle $VECM(1)$

On souhaite obtenir les coefficients du $VMA(2)$ associés au VECM.

```
# Coefficient de la représentation VMA infinie
ma_infty = vecm2.ma_rep(maxn=2)
ma = pd.DataFrame(np.append(ma_infty[0],
    np.append(ma_infty[1],ma_infty[2], axis=1).round(2),axis=1),
    columns = ["e1.L0","e2.L0","e3.L0","e1.L1","e2.L1","e3.L1","e1.L2",
        "e2.L2","e3.L2"],
    index = ["ngui.L0","ndah.L0", "fialah.L0"])
```

Table 4.26 – Coefficients de la forme VMA(2) du VECM(1)

	e1.L0	e2.L0	e3.L0	e1.L1	e2.L1	e3.L1	e1.L2	e2.L2	e3.L2
ngui.L0	1	0	0	0.92	0.02	0.20	0.19	0.65	0.58
ndah.L0	0	1	0	0.43	0.38	-0.26	0.31	0.61	-0.12
fialah.L0	0	0	1	0.54	-0.47	0.75	0.17	-0.06	1.03

Validation des modèles VECM

Sommaire

5.1 Diagnostic des résidus et prévision	120
5.2 Analyse structurelle d'un modèle VECM	126

5.1 Diagnostic des résidus et prévision

Le diagnostic des résidus est également une étape importante dans la modélisation des processus vectoriels à correction d'erreur. Plusieurs tests d'adéquation des modèles vus dans le cadre des processus VAR stationnaires peuvent être étendus aux processus vectoriels à correction d'erreur : test d'autocorrélation, test de normalité, etc.

5.1.1 Test d'autocorrélation des résidus

Afin d'étudier les propriétés des autocovariances et autocorrélations des résidus d'un processus VECM, nous notons les résidus estimés par $\hat{\varepsilon}_t$. Au chapitre (9), on a noté

$$\begin{aligned}\widehat{C}_i &:= \frac{1}{T} \sum_{t=i+1}^{t=T} \hat{\varepsilon}_t \hat{\varepsilon}_t' \quad i = 0, 1, \dots, h \\ \widehat{\mathbf{C}}_h &:= (\widehat{C}_1, \dots, \widehat{C}_h), \quad \widehat{\mathbf{c}}_h := \text{vec}(\widehat{\mathbf{C}}_h)\end{aligned}$$

les autocovariances des résidus et $\widehat{R}_i (i = 0, 1, \dots, h)$, $\widehat{\mathbf{R}}_h := (\widehat{R}_1, \dots, \widehat{R}_h)$ et $\widehat{\mathbf{r}}_h := \text{vec}(\widehat{\mathbf{R}}_h)$ les autocorrélations des résidus.

5.1.1.1 Test de Portmanteau

Brüggemann, Lütkepohl, and Saikkonen (2006) montre que le test de portmanteau et du multiplicateur de Lagrange pour l'autocorrélation des résidus peuvent être utilisés conjointement sur un processus VECM. Dans ce cas, la statistique suivante de portmanteau

$$Q_h := T \sum_{i=1}^{i=h} \text{trace}(\widehat{C}_i' \widehat{C}_0^{-1} \widehat{C}_i \widehat{C}_0^{-1}) = T \widehat{\mathbf{c}}_h' (I_h \otimes \widehat{C}_0^{-1} \otimes \widehat{C}_0^{-1}) \widehat{\mathbf{c}}_h \quad (5.1)$$

suit approximativement une loi de χ^2 à $(hd^2 - d^2(p-1) - dr)$ degrés de liberté.

Notons que cette statistique de portmanteau n'est valable que pour les échantillons de grande taille et pour des h qui tendent vers l'infini. Il n'est pas approprié pour des h petits. Ainsi, pour des petits échantillons la version modifiée du test de portmanteau est proposée :

$$\overline{Q}_h := T^2 \sum_{i=1}^{i=h} (T-i)^{-1} \text{trace}(\hat{C}_i' \hat{C}_0^{-1} \hat{C}_i \hat{C}_0^{-1}) \quad (5.2)$$

Sous H_0 , cette statistique \overline{Q}_h suit une loi de χ^2 à $(hd^2 - d^2(p-1) - dr)$ degrés de liberté.

Exemple 5.1 *Test de portmanteau sur les résidus estimés du VECM(1)*

Sous Python, le test de portmanteau multivarié d'un processus VECM est calculé grâce à la méthode `res.test_whiteness()`.

```
# Extraction des résultats du test
def whiteness_output(res):
    print("=="*35);print(res.title);print(res.signif_str);print("=="*35)
    print(res.h0);print("=="*35)
    df = pd.DataFrame({
        "Test statistic": res.test_statistic, "Critical value" : res.crit_value,
        "df" : res.df, "p-value" : res.pvalue, "signif" : res.signif},
        index = ["Portmanteau test"])
    print(df); print("=="*35); print(res.conclusion_str);print("=="*35)
# Test d'autocorrélation (Portmanteau test)
white_test = vecm2.test_whiteness(nlags=12,signif=0.05, adjusted=True)
whiteness_output(white_test)

## =====
## Adjusted Portmanteau-test for residual autocorrelation
## at 5% significance level
## =====
## H_0: residual autocorrelation up to lag 12 is zero
## =====
##              Test statistic  Critical value  df   p-value  signif
## Portmanteau test      95.329291      119.870939  96  0.500141   0.05
## =====
## Conclusion: fail to reject H_0
## =====
```

Nous effectuons le test de portmanteau des retards h allant de 2 à 12.

```
# Test de portmanteau pour h = 1,...,12 au seuil de 5%
pm_vecm = pd.DataFrame(np.zeros((3,11)),index=["test statistic","df","p-value"],
                        columns = [f"Q({x+2})" for x in range(11)])
for i in range(2,13):
    res = vecm2.test_whiteness(nlags=i,signif=0.05,adjusted=True)
    pm_vecm.iloc[:,i-2] = [res.test_statistic,res.df,res.pvalue]
```

Table 5.1 – Test de portmanteau pour h allant de 2 à 12

	Q(2)	Q(3)	Q(4)	Q(5)	Q(6)	Q(7)	Q(8)	Q(9)	Q(10)	Q(11)	Q(12)
test statistic	10.94	24.50	30.63	43.07	51.86	56.41	66.69	77.18	81.09	90.12	95.33
df	6.00	15.00	24.00	33.00	42.00	51.00	60.00	69.00	78.00	87.00	96.00
p-value	0.09	0.06	0.16	0.11	0.14	0.28	0.26	0.23	0.38	0.39	0.50

5.1.1.2 Test du multiplicateur de Lagrange : LM test

La distribution asymptotique de la statistique du multiplicateur de Lagrange (LM test) pour l'autocorrélation des résidus n'est pas affectée en présence des variables intégrées. Nous pouvons utiliser le modèle de régression auxiliaire suivant :

$$\hat{\varepsilon}_t = \alpha \hat{\beta}' X_{t-1} + \sum_{j=1}^{j=p-1} \Theta_j \Delta X_{t-j} + \sum_{i=1}^{i=h} D_i \hat{\varepsilon}_{t-i} + \eta_t, \quad t = 1, \dots, T \quad (5.3)$$

avec $\hat{\varepsilon}_s = 0$ pour $s < 1$. Les hypothèses du test sont :

$$\begin{cases} H_0 : D_1 = \dots = D_h = 0 \\ H_1 : \exists k \in \{1, \dots, h\} \text{ tel que } D_k \neq 0 \end{cases}$$

Sous l'hypothèse nulle, la statistique du multiplicateur de Lagrange suit approximativement une loi de χ^2 à (hd^2) .

5.1.2 Test de normalité

Le test de normalité est basé sur les résidus estimés du processus VAR. Cependant, nous pouvons utiliser les résidus estimés du modèle VECM sans pour autant affecter les propriétés asymptotiques des tests statistiques (voir Kilian and Demiroglu (2000)).

Exemple 5.2 Test de normalité sur les résidus estimés du VECM(1)

On calcule le test de normalité multivarié grâce à la méthode `res.test_normality()`. L'hypothèse nulle associée est celle de la normalité des résidus.

```
# Extraction des résultats du test causalité
def normality_output(res):
    title = res.title + res.signif_str
    print("====*24"); print(title); print("====*24"); print(res.h0); print("====*24")
    df = pd.DataFrame({
        "Test statistic": res.test_statistic, "Critical value" : res.crit_value,
        "df" : res.df, "p-value" : res.pvalue, "signif" : res.signif},
        index = ["Jarque - Bera test"])
    print(df); print("====*24"); print(res.conclusion_str); print("====*24")
# Test de normalité
norm_test = vecm2.test_normality(signif=0.05)
normality_output(norm_test)

## =====
## normality (skew and kurtosis) test at 5% significance level
```

```
## =====
## H_0: data generated by normally-distributed process
## =====
##              Test statistic  Critical value  df    p-value  signif
## Jarque - Bera test          7.770409      12.591587   6  0.255411   0.05
## =====
## Conclusion: fail to reject H_0
## =====
```

On a une pvalue qui n'est pas significative, donc on accepte l'hypothèse nulle de normalité des résidus.

5.1.3 Prévision

On considère le modèle défini par l'équation (4.11) où les variables sont intégrées, si possible cointégrées. Notons par $\hat{\Phi}_j(j = 1, \dots, p)$, les coefficients estimés et $\hat{\Sigma}_\varepsilon$, la matrice des variances covariances estimées des résidus. La prévision à l'ordre h est estimée par :

$$\hat{X}_t(h) = \hat{\Phi}_1 \hat{X}_t(h-1) + \dots + \hat{\Phi}_p \hat{X}_t(h-p) \quad (5.4)$$

où $\hat{X}_t(j) := X_{t+j}$ pour $j \leq 0$. L'erreur de prévision à la même forme que celle d'un VAR stable. Pour cela, écrivons le processus VAR(p) sous la forme d'un VAR(1) :

$$\underline{X}_t = \Omega \underline{X}_{t-1} + \varepsilon_t \quad (5.5)$$

Si ε_t est un bruit blanc indépendant, la prévision optimale à l'horizon h de \underline{X}_t est

$$\underline{X}_t(h) = \Omega \underline{X}_t(h-1) = \Omega^h \underline{X}_t$$

De plus,

$$\begin{aligned} \underline{X}_{t+h} &= \Omega \underline{X}_{t+(h-1)} + \varepsilon_t \\ &= \Omega^h \underline{X}_t + \varepsilon_t + \Omega \varepsilon_{t+(h-1)} + \dots + \Omega^h \varepsilon_{t+1} \end{aligned}$$

Ainsi, l'erreur de prévision devient :

$$\underline{X}_{t+h} - \hat{\underline{X}}_t(h) = \varepsilon_t + \Omega \varepsilon_{t+(h-1)} + \dots + \Omega^h \varepsilon_{t+1}$$

En multipliant cette équation par $T := [I_d : 0 : \dots : 0]$ de format $(d \times dp)$, on obtient :

$$\begin{aligned} X_{t+h} - X_t(h) &= T \varepsilon_t + T \Omega T' \varepsilon_{t+(h-1)} + \dots + T \Omega^{h-1} T' \varepsilon_{t+1} \\ &= \varepsilon_t + \Psi_1 \varepsilon_{t+(h-1)} + \dots + \Psi_{h-1} \varepsilon_{t+1} \end{aligned}$$

où $T' T \varepsilon_t = \varepsilon_t$ et $\Psi_i = T \Omega^i T'$.

Par ailleurs, les Ψ_i peuvent être obtenus à partir des $\Phi_{-}\{i\}$ par récurrence. En effet, on a la relation de récurrence suivante :

$$\begin{cases} \Psi_0 = I_d \\ \Psi_i = \sum_{j=1}^{i-1} \Psi_{i-j} \Phi_j, \quad i = 1, 2, \dots \end{cases} \quad (5.6)$$

Afin de construire des intervalles de confiance, la matrice minimal de prévision de X_t est donné par :

$$\Sigma_X(h) = \sum_{i=0}^{h-1} \Psi_i \Sigma_\varepsilon \Psi_i' \quad (5.7)$$

Cependant, il existe une différence vraiment important par rapport au processus VAR stationnaire. Dans le cas du VAR stable, la matrice Ψ_i converge vers zéro lorsque t tend l'infini et $\Sigma_X(h)$ converge vers la matrice des covariances de X_t lorsque h tend vers l'infini. Ce résultat est obtenu parce que les valeurs propres de la matrice Ω ont un module inférieur à un. Ainsi, $\Psi_i = T\Omega^i T'$ tend vers zéro quand i tend vers l'infini. Parce que les valeurs propres de Ω correspondent à l'inverse des racines du polynôme $\det(I_d - \Phi_1 z - \dots - \Phi_p z^p)$, Ψ_i ne converge pas vers zéro dans le cas des processus instables où une ou plusieurs valeurs propres sont 1. Par conséquent, certains éléments de la matrice $\Sigma_X(h)$ seront proche de l'infini lorsque h tendra vers l'infini.

Exemple 5.3 Illustration

Considérons le processus VAR(1) bivarié avec une relation de cointégration :

$$\begin{bmatrix} x_{1,t} \\ x_{2,t} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_{1,t-1} \\ x_{2,t-1} \end{bmatrix} + \begin{bmatrix} \varepsilon_{1,t} \\ \varepsilon_{2,t} \end{bmatrix} \quad (5.8)$$

La forme VECM associé est :

$$\begin{aligned} \Delta X_t &= -(I - \Phi_1) X_{t-1} + \varepsilon_t \\ \begin{bmatrix} x_{1,t} \\ x_{2,t} \end{bmatrix} &= - \begin{bmatrix} 1 & -1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_{1,t-1} \\ x_{2,t-1} \end{bmatrix} + \begin{bmatrix} \varepsilon_{1,t} \\ \varepsilon_{2,t} \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \end{bmatrix} \begin{bmatrix} 1 & -1 \end{bmatrix} \begin{bmatrix} x_{1,t-1} \\ x_{2,t-1} \end{bmatrix} + \begin{bmatrix} \varepsilon_{1,t} \\ \varepsilon_{2,t} \end{bmatrix} \end{aligned}$$

Par identification, on a : $\alpha = \begin{bmatrix} -1 \\ 0 \end{bmatrix}$ et $\beta = \begin{bmatrix} 1 & -1 \end{bmatrix}$.

Pour ce processus, il est facile de montrer que $\Psi_0 = I_2$ et $\Psi_j = \Phi_1^j = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}$ $j = 1, 2, \dots$. Ce qui implique

$$\Sigma_X(h) = \sum_{j=0}^{h-1} \Psi_j \Sigma_\varepsilon \Psi_j = \Sigma_\varepsilon + (h-1) \begin{bmatrix} \sigma_2^2 & \sigma_2^2 \\ \sigma_2^2 & \sigma_2^2 \end{bmatrix}$$

où σ_2^2 est la variance de $\varepsilon_{2,t}$. En outre, l'espérance conditionnelle est $x_{k,t}(h) = x_{2,t}$ ($k = 1, 2$). Ainsi, l'intervalle de confiance de prévision est :

$$\left[x_{2,t} - z_{\alpha/2} \sqrt{\sigma_k^2 + (h-1)\sigma_2^2}; x_{2,t} + z_{\alpha/2} \sqrt{\sigma_k^2 + (h-1)\sigma_2^2} \right] \quad k = 1, 2$$

où $z_{\alpha/2}$ est le quantile d'ordre $(1 - \alpha/2)$ de la loi normale centrée réduite. Il est facile de voir que la longueur de cette intervalle est non bornée lorsque h tend vers l'infini.

Notons par $\hat{X}_t(h)$ la prévision à l'horizon h de X_t , alors on a :

$$\hat{X}_t(h) = \hat{\Phi}_1 \hat{X}_t(h-1) + \dots + \hat{\Phi}_p \hat{X}_t(h-p) \quad (5.9)$$

où $\hat{X}_t(i) = X_{t+i}$ pour $i \leq 0$. Pour cette prédicteur, l'erreur de prévision devient :

$$X_{t+h} - \hat{X}_t(h) = [X_{t+h} - X_t(h)] + [X_t(h) - \hat{X}_t(h)]$$

Ce qui équivaut à :

$$X_{t+h} - \hat{X}_t(h) = \sum_{i=0}^{h-1} \Psi_i \varepsilon_{t+h-i} + [X_t(h) - \hat{X}_t(h)] \quad (5.10)$$

Dans le cas des processus stationnaires et sous l'hypothèse de normalité, le second terme de l'équation (5.10) converge en probabilité vers zéro, $X_t(h) - \hat{X}_t(h) = o_p(1)$. Cependant, dans le cas présent, cela est difficile à justifier (cf. Basu and Roy (1987)). L'estimateur MSE de $\Sigma_X(h)$ est :

$$\hat{\Sigma}_X(h) = \sum_{i=0}^{h-1} \hat{\Psi}_i \hat{\Sigma}_\varepsilon \hat{\Psi}_i' \quad (5.11)$$

où $\hat{\Psi}_i$ peut être obtenu à partir des coefficients estimés $\hat{\Phi}_i$ par récurrence.

Exemple 5.4 Prévision à partir du VECM(1)

Sous Python, la prévision s'effectue avec la méthode `res.predict()` en spécifiant la longueur h de prévision et le seuil de significativité α pour les intervalles de confiance. Par défaut, le paramètre α est fixé à 5%.

```
# Prévision à horizon h = 6
vecm_pred = vecm2.predict(steps=6, alpha=0.05)
ngui_pred = pd.DataFrame(vecm_pred[0],
                          columns = ["ngui.mean", "ngui.ci.lo", "ngui.ci.up"])
ndah_pred = pd.DataFrame(vecm_pred[1],
                          columns = ["ndah.mean", "ndah.ci.lo", "ndah.ci.up"])
fialah_pred = pd.DataFrame(vecm_pred[2],
                           columns = ["fialah.mean", "fialah.ci.lo", "fialah.ci.up"])
pred = pd.concat([ngui_pred, ndah_pred, fialah_pred], axis=1)
pred.index = pd.date_range('2020-07-30', periods=len(pred), freq = "M")
```

Table 5.2 – Prévision du VECM(1) pour h=6

	Prévision sur ngui			Prévision sur ndah			Prévision sur fialah		
	ngui.mean	ngui.ci.lo	ngui.ci.up	ndah.mean	ndah.ci.lo	ndah.ci.up	fialah.mean	fialah.ci.lo	fialah.ci.up
2020-07-31	26.26	24.44	33.11	23.10	21.73	30.26	29.42	27.16	35.96
2020-08-31	24.69	23.98	32.12	20.20	20.58	28.42	29.18	27.38	35.82
2020-09-30	24.45	23.91	31.74	18.82	19.77	26.92	30.08	28.06	36.55
2020-10-31	24.78	23.80	31.74	18.35	19.11	25.97	31.20	28.50	37.50
2020-11-30	25.11	23.87	31.96	18.08	18.68	25.41	32.14	29.07	38.51
2020-12-31	25.13	23.91	32.05	17.57	18.28	24.87	32.69	29.54	39.22

```
# Représentation graphique
vecm2.plot_forecast(steps=6,alpha=0.05,plot_conf_int=True)
plt.show()
```

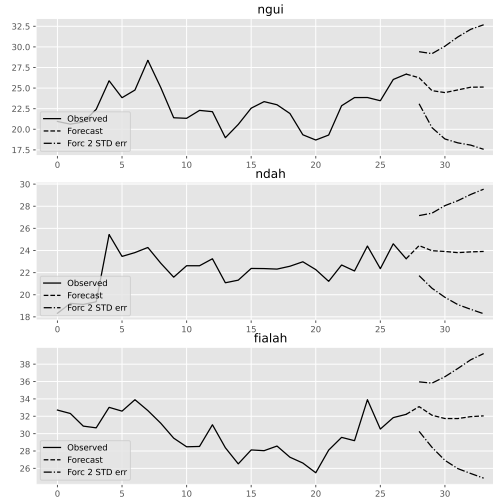


Figure 5.1 – Prédiction du VECM(1) pour les mois de juillet 2020 à décembre 2020

5.2 Analyse structurelle d'un modèle VECM

L'analyse structurelle dans le cadre du modèle vectoriel à correction d'erreur est identique à celle vue dans les modèles VAR : causalité au sens de Granger, causalité instantannée, fonction de réponses impulsionnelles. Les interprétations des résultats sont à prendre avec une certaine précaution .

5.2.1 Causalité au sens de Granger

Supposons que le processus X_t d -dimensionnel associé au modèle VAR(p) soit partitionné en deux sous vecteurs Y_t et Z_t de dimension d_1 et $d - d_1$ respectivement, c'est - à - dire :

$$X_t = \begin{bmatrix} Y_t \\ Z_t \end{bmatrix} \quad \text{et} \quad \Phi_j = \begin{bmatrix} \Phi_{11}^j & \Phi_{12}^j \\ \Phi_{21}^j & \Phi_{22}^j \end{bmatrix}, \quad j = 1, \dots, p$$

où Φ_i est partitionné en accord avec la partition de X_t . Alors Z_t ne cause pas au sens de Granger Y_t si et seulement si $\Phi_{12}^j = 0$, $j = 1, \dots, p$. Y_t ne cause pas au sens de Granger Z_t si et seulement si $\Phi_{21}^j = 0$, $j = 1, \dots, p$.

Cette approche par partitionnement est également possible dans l'étude des processus VECM. En effet, on a la partition suivante de ΔX_t :

$$\begin{bmatrix} \Delta Y_t \\ \Delta Z_t \end{bmatrix} = \begin{bmatrix} \pi_{11} & \pi_{12} \\ \pi_{21} & \pi_{22} \end{bmatrix} \begin{bmatrix} Y_{t-1} \\ Z_{t-1} \end{bmatrix} + \sum_{j=1}^{p-1} \begin{bmatrix} \Theta_{11}^j & \Theta_{12}^j \\ \Theta_{21}^j & \Theta_{22}^j \end{bmatrix} \begin{bmatrix} \Delta Y_{t-j} \\ \Delta Z_{t-j} \end{bmatrix} + \varepsilon_t$$

Il s'en suit que $\Phi_{12}^j = 0$ est équivalent à

$$\pi_{12} = 0 \quad \text{et} \quad \Theta_{12}^j = 0 \quad j = 1, \dots, p-1 \quad (5.12)$$

Proposition 5.1 *Test de Wald de mesure de causalité au sens de Granger*

Le test de Wald de causalité consiste à tester les hypothèses suivantes :

$$H_0 : R\beta = 0 \quad \text{contre} \quad H_0 : R\beta \neq 0 \quad (5.13)$$

où $\beta = \text{vec}[\Phi_1, \dots, \Phi_p]$ et R une matrice de format $(N \times pd^2)$ de rang N . La statistique de Wald est définie par :

$$\lambda_W = T\hat{\beta}' R' (R\hat{\Sigma}_\beta^{co} R')^{-1} R\hat{\beta} \quad (5.14)$$

où $\hat{\Sigma}_\beta^{co}$ est l'estimateur consistant de Σ_β^{co} .

Sous H_0 , cette statistique λ_W suit une distribution de χ^2 à N degrés de liberté et $\text{rang}(R\hat{\Sigma}_\beta^{co} R') = N$.

Cette proposition est vraie dans le cas d'un VAR stable (r relations de cointégration). Or le nombre de relation de cointégration, r , peut être inférieur à d , dans ce cas la matrice Σ_β^{co} est singulière ($\text{rang}(R\hat{\Sigma}_\beta^{co} R') < N$). Une distribution limite du χ^2 pour λ_W peut être obtenue si l'inverse de $R\hat{\Sigma}_\beta^{co} R'$ est remplacé par l'inverse généralisée. Dans ce cas, sous H_0 , λ_W suit une distribution asymptotique χ^2 à $\text{rang}(R\Sigma_\beta^{co} R')$ si

$$\text{rang}(R\hat{\Sigma}_\beta^{co} R') = \text{rang}(R\Sigma_\beta^{co} R') \quad (5.15)$$

avec probabilité égale à 1 (*cf.* Andrews (1987)).

Andrews (1987) montre que sous certaines conditions, la distribution asymptotique de λ_W peut ne pas être une χ^2 . Plusieurs critiques ont été soulevés par rapport à cette statistique de Wald dans le cadre des systèmes cointégrés (*cf.* Toda and Yamamoto (1995), Dolado and Lütkepohl (1996)).

Proposition 5.2 *Test de Fisher de mesure de causalité au sens de Granger*

Cette distribution asymptotique du χ^2 est faible pour des échantillons de petite taille. Par conséquent, on peut ajuster la statistique de Wald et obtenir une nouvelle statistique λ_F définie par :

$$\lambda_F = \lambda_W / N \quad (5.16)$$

On comparera donc ces statistiques à leurs valeurs critiques afin de rejeter ou non l'hypothèse nulle de non - causalité au sens de Granger qui est testée.

Exemple 5.5 *Causalité au sens de Granger*

Pour mesurer la causalité au sens de Granger, on fait appel à la méthode `model.test_granger_causality()` en spécifiant la variable dont on voudrait mesurer la causalité. On rejette H_0 si la p-value est inférieure au seuil de significativité (généralement fixé à 5%).

— H_0 : (ndah, fialah) ne cause pas au sens de granger ngui

```
# causalité de Granger sur ngui
ngui_f = vecm2.test_granger_causality(caused="ngui",causing=["ndah","fialah"])
granger_fisher_output(ngui_f)

## =====
## Granger causality F-test at 5% significance level
## =====
## H_0: ['ndah', 'fialah'] do not Granger-cause ngui
## =====
##          Test statistic  Critical value  ddl1  ddl2  p-value  signif
## granger          4.588322          2.553395    4    51  0.003055    0.05
## =====
## Conclusion: reject H_0
## =====
```

La pvalue est significative au seuil de 5%, par conséquent on rejette l'hypothèse nulle. Donc (ndah, fialah) cause au sens de granger ngui.

— H_0 : (ngui, fialah) ne cause pas au sens de granger ndah

```
# causalité de Granger sur ndah
gndah_f = vecm2.test_granger_causality(caused="ndah",causing=["ngui","fialah"])
granger_fisher_output(gndah_f)

## =====
## Granger causality F-test at 5% significance level
## =====
## H_0: ['ngui', 'fialah'] do not Granger-cause ndah
## =====
##          Test statistic  Critical value  ddl1  ddl2  p-value  signif
## granger          1.818375          2.553395    4    51  0.139636    0.05
## =====
## Conclusion: fail to reject H_0
## =====
```

La pvalue n'est pas significative au seuil de 5%, on accepte l'hypothèse nulle : (ngui, fialah) ne cause pas au sens de granger ndah.

— H_0 : (ngui, ndah) ne cause pas au sens de granger fialah

```
# causalité de Granger sur fialah
gfialah_f = vecm2.test_granger_causality(caused="fialah",causing=["ngui","ndah"])
granger_fisher_output(gfialah_f)

## =====
## Granger causality F-test at 5% significance level
## =====
## H_0: ['ngui', 'ndah'] do not Granger-cause fialah
## =====
```

```
##          Test statistic Critical value  ddl1  ddl2  p-value  signif
## granger          2.011733          2.553395    4    51  0.106737    0.05
## =====
## Conclusion: fail to reject H_0
## =====
```

La pvalue n'est pas significative au seuil de 5%, on accepte l'hypothèse nulle : (ngui, ndah) ne cause pas au sens de granger fialah.

On effectue le test de causalité au sens de Granger pour chacune des variables prises deux à deux.

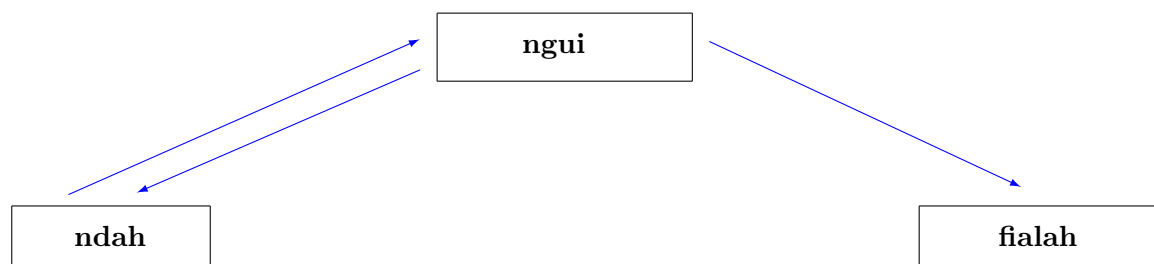
```
def grangers_causation_matrix2(model,**kwargs):
    stat_pvalue = dict()
    for r in model.names:
        for c in model.names:
            res = model.test_granger_causality(caused=r,causing=c,**kwargs)
            stat_pvalue[(r,c)] = (res.test_statistic, res.pvalue)
    df = pd.DataFrame(stat_pvalue,index = ["statistic", "p-value"]).T.round(4)
    df = (df.reset_index()
          .rename(columns={"level_0": "caused", "level_1": "causing"}))
    return df
# Application
granger_vecm = grangers_causation_matrix2(vecm2)
```

Table 5.3 – Test de causalité de Granger

caused	causing	statistic	p-value
ngui	ngui	12.5152	0.0000
ngui	ndah	5.8662	0.0051
ngui	fialah	3.6599	0.0327
ndah	ngui	3.4617	0.0389
ndah	ndah	2.5777	0.0858
ndah	fialah	0.8283	0.4426
fialah	ngui	2.7381	0.0742
fialah	ndah	2.8787	0.0654
fialah	fialah	6.2862	0.0036

Le tableau (5.3) montre qu'on a 3 causalités au sens de Granger qui sont significatives.

Nous pouvons résumer le test de causalité de Granger entre les variables **ngui**, **ndah** et **fialah** en faisant un schéma pour montrer clairement le sens de causalité entre elles.



Causalité au sens de Granger ngui/ndah/fialah

5.2.2 Causalité de Granger au sens de Toda - Yamamoto

Plusieurs critiques, à placer au passif des tests de causalité traditionnels (principalement celui de Granger), consacrent l'efficacité du test de causalité de Granger au sens de Toda and Yamamoto (1995). On rappelle que le test de causalité au sens de Granger ne s'applique que sur des séries stationnaires ou (stationnarisées), ce qui rend indispensable les tests préliminaires de cointégration des séries ou l'analyse de leur stationnarité avant de vérifier une éventuelle causalité entre elles.

Pourtant, les tests de racine unitaire sont moins efficaces sur des petites échantillons et ne sont pas toujours et ne sont pas toujours sans biais. Aussi, en procédant à la transformation des séries par la différence première, par souci de stationnarisation ou de cointégration, l'on obtient des bonnes propriétés statistiques tout en perdant de l'information sur le niveau des séries, laquelle information en niveau ne devrait pas être supprimée vu qu'elle est enrichissante pour expliquer la dynamique du modèle étudiée.

Il suit que, sur des petits échantillons, le test de cointégration de Johansen est sensible à certains paramètres de choix qui sont de nature à l'affaiblir : le lag ou décalage (risque d'estimer un VAR sous paramétré) et la présence (ou absence) de tendance déterministe dans le VAR et/ou l'espace de cointégration (risque de perte en degré de liberté). Ces paramètres créent des biais qui amènent souvent à rejeter l'hypothèse d'absence de cointégration (H_0) alors qu'elle est vraie.

Cette faiblesse des résultats de cointégration, couplée au caractère biaisé des tests de racine unitaire, réduisent l'efficacité du test de causalité de Granger (issue aléatoire) et poussent Toda and Yamamoto (1995) à proposer des procédures non séquentielles pour tester la causalité entre séries. Pour ces auteurs, les tests préliminaires de stationnarité et cointégration (procédures séquentielles de Granger) importent peu pour l'économiste qui doit se soucier à tester plutôt les restrictions théoriques (elles sécurisent l'information en niveau).

Ces deux auteurs vont proposer un VAR en niveau corrigé (sur - paramétré), devant servir de base au test de causalité, sous l'hypothèse d'une potentielle cointégration probable entre séries qu'ils intèrent dans le modèle sans l'étudier comme tel (explicitement). La procédure du test de causalité de Granger proposée par Toda and Yamamoto (1995) est la suivante :

- Trouver l'ordre d'intégration maximale des séries sous études (d) en recourant aux tests de racine unitaire
- Déterminer la retard optimal (p) du VAR en niveau sous étude en recourant aux critères d'information (AIC, BIC, HQ).
- Estimer un VAR en niveau augmenté (sur - paramétré) d'ordre ($p + d$)

En ce qui concerne l'estimation du VAR en niveau augmenté, les conditions de stationnarité des séries définiront le retard à ajouter au VAR en niveau. Au fait, pour des séries stationnaires en niveau, aucun retard n'est ajouté au VAR (procédure de test standard) ; par contre, pour des séries $I(1)$, l'on ajoutera un retard au VAR, ainsi de suite.

Exemple 5.6 Exemple illustratif de l'approche de Toda - Yamamoto

Si l'on veut tester la causalité entre deux séries $X_{1,t}$ et $X_{2,t}$ au sens de Toda et Yamamoto, l'on devra estimer le VAR augmenté comme suit :

$$\begin{cases} X_{1,t} = \phi_0^{(1)} + \sum_{k=1}^{k=p} \phi_{1,k}^{(1)} X_{1,t-k} + \sum_{k=p+1}^{k=d} \psi_{1,k}^{(1)} X_{1,t-k} + \sum_{k=1}^{k=p} \phi_{2,k}^{(1)} X_{2,t-k} + \sum_{k=p+1}^{k=d} \psi_{2,k}^{(1)} X_{2,t-k} + \varepsilon_{1,t} \\ X_{2,t} = \phi_0^{(2)} + \sum_{k=1}^{k=p} \phi_{1,k}^{(2)} X_{1,t-k} + \sum_{k=p+1}^{k=d} \psi_{1,k}^{(2)} X_{1,t-k} + \sum_{k=1}^{k=p} \phi_{2,k}^{(2)} X_{2,t-k} + \sum_{k=p+1}^{k=d} \psi_{2,k}^{(2)} X_{2,t-k} + \varepsilon_{2,t} \end{cases} \quad (5.17)$$

Le test de causalité sur un test VAR augmenté (ou sur - paramétré) va consister à tester des restrictions sur les d premiers coefficients, les autres paramètres étant nuls (ils traduisent une cointégration probable entre séries dans le modèle VAR). Ainsi, au sens de Toda et Yamamoto, les hypothèses de test sont (le test est basé sur la statistique W de Wald qui est distribuées suivant un χ^2 à n degrés de liberté, avec n le nombre de restrictions. Cete statistique est indépendante de l'ordre d'intégration des séries et leur cointégration) :

$$\begin{cases} H_0 : \phi_{2,1}^{(1)} = \dots = \psi_{2,p}^{(1)} = 0 & (\chi_{cal}^2 < \chi_{tab}^2; p - value \chi^2 > 5\%) : X_{2,t} \text{ ne cause pas } X_{1,t} \\ H_0 : \phi_{2,1}^{(2)} = \dots = \psi_{2,p}^{(2)} = 0 & (\chi_{cal}^2 < \chi_{tab}^2; p - value \chi^2 > 5\%) : X_{1,t} \text{ ne cause pas } X_{2,t} \end{cases}$$

L'on doit veiller à ce que l'ordre d'intégration maximale d ne dépasse par le retard optimal p du polynôme AR du VAR pour appliquer ce test.

Exemple 5.7 Test de causalité de Granger au sens de Toda - Yamamoto

On souhaite appliquer le test de causalité de Granger au sens de Toda - Yamamoto sur les séries de taux d'intérêt à court terme des villages NGUI, NDAH et FIALAH dans le groupement FONGO - NDENG.

Sous Python, il n'existe pas encore de fonction susceptible de réaliser le test de causalité de Granger au sens de Toda - Yamamoto. Cependant, en se basant sur la théorie évoquée ci - dessus, nous pouvons réaliser ce test en deux étapes :

- Estimer un modèle VAR(p) en tenant compte des retards allant de $p + 1$ à $p + d$ comme exogènes au modèle;
- Réaliser le test de causalité de Granger à partir du modèle estimé.

```
# Estimation du VAR en niveau augmenté
var_model=VAR(joh_df,exog =joh_df.shift(3).fillna(0)).fit(2,trend = "c")
```

On resume l'ensemble des résultats du test de causalité de Granger au sens de Toda 6 Yamamoto dans un tableau contenant les p-values.

```
granger_var2 = grangers_causation_matrix(var_model)
```

Le tableau 5.4 montre qu'on a 1 causalité de granger au sens de Toda - Yamamoto qui est significative.



Causalité de Granger au sens de Toda - Yamamoto entre ngui/ndah/fialah

5.2.3 Causalité instantanée

L'analyse de la causalité instantanée dans les systèmes intégrés ou cointégrés est quasi identique à celle des processus multivariés stationnaires. Cette analyse se base sur la matrice estimée des

Table 5.4 – Test de causalité de Granger au sens de Toda - Yamamoto

caused	causing	statistic	p-value
ngui	ngui	8.6375	0.0006
ngui	ndah	4.0464	0.0230
ngui	fialah	2.6110	0.0827
ndah	ngui	2.1282	0.1289
ndah	ndah	1.6527	0.2011
ndah	fialah	0.6247	0.5393
fialah	ngui	1.7460	0.1842
fialah	ndah	1.8491	0.1672
fialah	fialah	4.0833	0.0223

covariances des résidus $\widehat{\Sigma}_\varepsilon$.

Exemple 5.8 Causalité instantanée

Pour mesurer la causalité instantané, on fait appel à la méthode `model.test_inst_causality()` en spécifiant la variable dont on voudrait mesurer la causalité. On rejette H_0 si la p-value est inférieure au seuil de significativité (généralement fixé à 5%).

— H_0 : ngui ne cause pas instantanément (ndah,fialah)

```
# H_0 : ngui ne cause pas instantanément (ndah,fialah)
ngui_inst_test = vecm2.test_inst_causality(causing = "ngui", signif = 0.05)
instantaneous_output(ngui_inst_test)
```

```
## =====
## Instantaneous causality Wald-test at 5% significance level
## =====
## H_0: ngui does not instantaneously cause [ndah, fialah]
## =====
##          Test statistic  Critical value  df   p-value  signif
## inst          9.930948         5.991465   2  0.006975   0.05
## =====
## Conclusion: reject H_0
## =====
```

La pvalue est significative au seuil de 5%, par conséquent on rejette l'hypothèse nulle : ngui cause instantanément (ndah,fialah).

— H_0 : ndah ne cause pas instantanément (ngui,fialah)

```
# H_0 : ndah ne cause pas instantanément (ngui,fialah)
ndah_inst_test = vecm2.test_inst_causality(causing = "ndah", signif = 0.05)
instantaneous_output(ndah_inst_test)
```

```
## =====
## Instantaneous causality Wald-test at 5% significance level
## =====
## H_0: ndah does not instantaneously cause [ngui, fialah]
## =====
##          Test statistic  Critical value  df   p-value  signif
```

```
## inst      10.737305      5.991465    2  0.00466    0.05
## =====
## Conclusion: reject H_0
## =====
```

La pvalue est significative au seuil de 5%, par conséquent on rejette l'hypothèse nulle : **ndah** cause instantanément (**ngui,fialah**).

— H_0 : **fialah** ne cause pas instantanément (**ngui,ndah**)

```
# H_0 : fialah ne cause pas instantanément (ngui, ndah)
fialah_inst_test = vecm2.test_inst_causality(causing = "fialah", signif = 0.05)
instantaneous_output(fialah_inst_test)
```

```
## =====
## Instantaneous causality Wald-test at 5% significance level
## =====
## H_0: fialah does not instantaneously cause [ngui, ndah]
## =====
##      Test statistic Critical value df  p-value signif
## inst      6.097948      5.991465  2  0.047408    0.05
## =====
## Conclusion: reject H_0
## =====
```

La pvalue est significative au seuil de 5%, par conséquent on rejette l'hypothèse nulle : **fialah** cause instantanément (**ngui,ndah**).

Pour mesurer la causalité instantanée des différentes variables prises deux à deux, on affiche la matrice de covariance des résidus :

```
# Matrice de covariance des résidus
print(vecm2.sigma_u)

## [[2.59932858 1.76633173 0.82840567]
##  [1.76633173 1.91911655 0.99653766]
##  [0.82840567 0.99653766 2.11367194]]
```

5.2.4 Analyse des réponses impulsionnelles

En principe, l'analyse des réponses impulsionnelles dans les systèmes intégrés et cointégrés peuvent être conduite de la même manière que dans le cadre des systèmes stationnaires. Cependant, elle doit être interprétée avec prudence. Dans un système cointégré, le terme $\beta' X_t$ est généralement considéré comme représentant la relation d'équilibre de long terme entre les variables. Supposons qu'il y a juste une relation de cointégration

$$\beta_1 x_{1,t} + \dots + \beta_d x_{d,t} = 0$$

or, si $\beta_1 \neq 0$,

$$x_{1,t} = -\frac{\beta_2}{\beta_1} x_{2,t} - \dots - \frac{\beta_d}{\beta_1} x_{d,t}$$

Cette relation montre qu'un accroissement d'une unité de $x_{2,t}$ a un effet long terme de β_2/β_1 sur $x_{1,t}$, ceteris paribus. Une innovation sur $x_{2,t}$ peut affecter plusieurs autres variables qui ont également un impact sur $x_{1,t}$. Par conséquent, l'effet long terme d'une $x_{2,t}$ -innovation sur $x_{1,t}$ peut être tout à fait différent de $-\beta_2/\beta_1$. Les réponses impulsionnelles peuvent donner une meilleure image des relations entre les variables.

Exemple 5.9 *Analyse des réponses impulsionnelles du VECM(1)*

Pour visualiser les réponses impulsionnelles, on fait appel à la méthode `res.irf()` en spécifiant le nombre de période sur laquelle on désire faire l'analyse. Pour notre exemple, nous fixons le paramètre à 5.

```
# Analyse des réponses impulsionnelles
irf = vecm2.irf(periods=5)
irf.plot(plot_stderr=False);
plt.show()
```

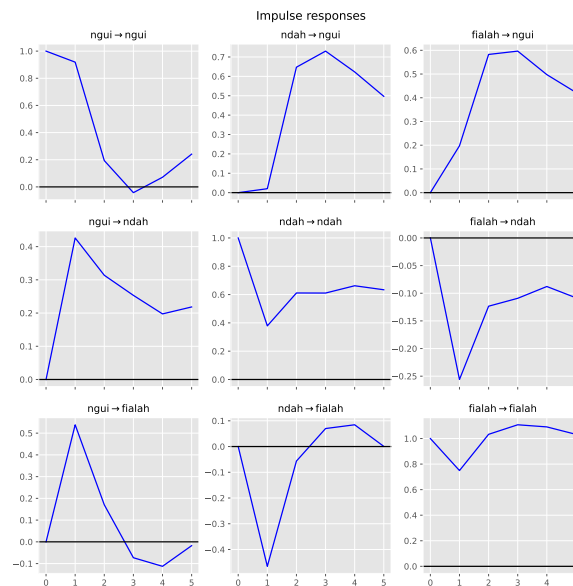


Figure 5.2 – Erreur de prévision des réponses impulsionnelles du VECM(1)

En conclusion, on soulignera que l'intérêt de la théorie de la cointégration est qu'elle fournit une méthode d'analyse des séries temporelles non stationnaires en évitant le problème de régressions fallacieuses. De plus, grâce aux modèles à correction d'erreur, elle permet de modéliser simultanément les dynamiques de long terme et de court terme des séries temporelles.

Modélisation ARDL et test de cointégration aux bornes

Sommaire

6.1 Modélisation ARDL	135
6.2 Test de cointégration de Pesaran et al.(2001)	141

Dans certaines spécifications des modèles économétriques temporelles, nous pouvons postuler que la variable endogène dépend non seulement de ses propres valeurs passées, mais également des valeurs prises par des variables exogènes à des époques antérieures : on parle alors de modèles autorégressifs à retards échelonnés (ARDL, *Autoregressive Distributed Lag*). Ils font partie de la classe des modèles dynamiques. Ces derniers ont la particularité de prendre en compte la dynamique temporelle (délai d'ajustement, anticipations, etc.) dans l'explication d'une chronique, améliorant ainsi les prévisions et l'efficacité des politiques publiques, contrairement au modèle non dynamique dont l'explication instantanée (effet immédiat ou non étalé dans le temps) ne restitue qu'une partie de la variation de la variable endogène.

Le modèle ARDL permet également de mener une étude sur l'existence d'une relation de cointégration ou de long terme (*cf.* Pesaran, Shin, and Smith (2001)). On appelle cette approche le test aux bornes (*bounds test*). Cette méthode résout la contrainte liée à l'ordre d'intégration que la méthode de Søren Johansen (1988) n'arrive pas. Cette dernière impose que toutes les variables doivent être intégrés du même ordre, soit 1. Ce qui n'est pas forcément le cas dans le test aux bornes.

6.1 Modélisation ARDL

6.1.1 Présentation générale

Le modèle ARDL « *Autoregressive Distributed Lag* » est une combinaison entre les modèles autorégressifs AR (*Autoregressive*) et les modèles à retards échelonnés DF (*Distributed Lag*). Sa forme générale s'écrit comme suit :

$$Y_t = c + \overbrace{\sum_{i=1}^p \alpha_i Y_{t-i}}^{AR} + \overbrace{\sum_{j=1}^k \sum_{l=0}^{q_j} \beta_{j,l} X_{j,t-l}}^{DL} + \varepsilon_t \quad (6.1)$$

où $Y_t, X_{1,t}, \dots, X_{k,t}$ et ε_t sont respectivement la variable endogène, les k variables exogènes et le

terme d'erreur.

Dans cette relation, le coefficient $\beta_{j,0}$ ($j = 1, \dots, k$) explique l'effet à court terme de $X_{j,t}$ sur Y_t .

Si l'on considère la relation de long terme ou d'équilibre suivante « $Y_t = \mu + \sum_{j=1}^{j=k} \lambda_j X_{j,t} + \eta_t$ », l'on peut calculer l'effet à long terme de $X_{j,t}$ sur Y_t (soit « λ_j ($j = 1, \dots, k$) ») comme suit :

$$\lambda_j = \frac{\sum_i \alpha_i}{1 - \sum_l \beta_{j,l}} \quad (6.2)$$

Le modèle $ARDL(p, q_1, \dots, q_k)$ peut s'écrire de façon plus concise en utilisant $k + 1$ polynômes retards : un pour les retards sur la variable dépendante Y (polynôme autorégressif) et les k autres pour les retards des variables explicatives.

Définition 6.1 *Le modèle $ARDL(p, q_1, \dots, q_k)$ est défini par la relation*

$$\Phi(B)Y_t = + \sum_{j=1}^{j=k} \Theta_j X_{j,t} + \varepsilon_t \quad (6.3)$$

où ε_t est un bruit blanc faible. Le polynôme $\Phi(B)$ vérifie $\Phi(B) = 1 - \sum_{i=1}^{i=p} \alpha_i B^i$, avec $\alpha_p \neq 0$.

On suppose que les racines du polynôme $\Phi(B)$ sont toutes situées en dehors du cercle unité. Le polynôme $\Theta_j(B)$ est défini par $\Theta_j(B) = \sum_{l=0}^{l=q_j} \beta_{j,l} B^l$ avec $j = 1, \dots, k$ et $\beta_{j,q_j} \neq 0$.

Nous pouvons réécrire notre modèle ARDL sous la forme d'un modèle à retards échelonnés contraints par inversion du polynôme $\Phi(B)$:

$$Y_t = \frac{c}{\Phi(1)} + \sum_{j=1}^k \frac{\Theta_j(B)}{\Phi(B)} X_{j,t} + \frac{\varepsilon_t}{\Phi(B)} = \mu + \sum_{j=1}^{j=k} \Psi_j(B) X_{j,t} + \eta_t \quad (6.4)$$

avec $\mu = c/\Phi(1)$, $\Psi_j = \Theta_j/\Phi(B)$ et $\eta_t = \varepsilon_t/\Phi(B)$. La détermination des termes des polynômes $\Psi_j(B)$ ($j = 1, \dots, k$) suppose d'inverser le polynôme $\Phi(B)$. Pour cela, il existe différentes méthodes.

Dans cette formulation, l'effet cumulatif de long terme de X_j sur Y est égal à

$$\Psi_j(1) = \sum_{i=1}^{\infty} \psi_{j,i} = \frac{\Theta_j(1)}{\Phi(1)} \quad (6.5)$$

La condition selon laquelle les racines du polynôme $\Phi(B)$ sont toutes situées en dehors du cercle unité garantit que l'effet dynamique (effet cumulé de long terme) de X_j sur Y est non explosif. Il convient de bien vérifier cette condition sur les modèles estimés sous peine d'obtenir des effets dynamiques non conformes à la réalité économique. Dans le cas où cette condition n'est pas vérifiée, il convient de différencier la variable Y_t et d'appliquer un nouveau modèle ARDL sur ΔY_t .

6.1.2 Estimation des paramètres d'un modèle ARDL

Les modèles ARDL souffrent généralement de problèmes d'autocorrélation d'erreurs avec la présence de la variable endogène décalée comme explicative, et de multicollinéarité, ce qui complique l'estimation des paramètres par la méthode des moindres carrés ordinaires. Ici, il tient de recourir aux techniques d'estimation robuste (méthode SUR, méthode des moments, etc.). Aussi, l'on retiendra que les variables considérées dans ces modèles se doivent d'être stationnaires pour éviter des régressions fallacieuses.

Exemple 6.1 Application du modèle ARDL

Nous illustrons notre exemple en utilisant les données [Danish Money Demand Data](#) disponible dans la librairie Statsmodels de Python. Ce jeu de données contient 55 observations et 5 variables : `lrm`, logarithme de la demande réelle de monnaie, `lry`, le logarithme du revenu réel, `lpy`, le logarithme du prix, `ibo`, le taux obligataire et `ide`, le taux de dépôt. Nous faisons abstraction de la variable `lpy` dans la modélisation.

```
# Chargement des données
from statsmodels.datasets import danish_data
data = danish_data.load_pandas().data
danish = data[["lrm", "lry", "ibo", "ide"]]
```

Sous Python, la fonction `ARDL` de la librairie Statsmodels permet d'effectuer une modélisation ARDL. On affecte la valeur 1 à tous les retards.

```
# Estimation des paramètres
from statsmodels.tsa.api import ARDL
lrm, exog = danish.lrm, danish[["lry", "ibo", "ide"]]
ardl = ARDL(endog=lrm, lags=1, exog=exog, order=1).fit()
ardl_params = extractParams(ardl, "lm")
```

On vérifie si notre unique racine a un module situé à l'extérieur du cercle unité :

```
# Module associé
print(abs(ardl.roots))
```

```
## [1.44594773]
```

La partie AR du modèle est stationnaire. On affiche les coefficients obtenus :

Table 6.1 – Coefficients du modèle ARDL(1,1,1,1)

	coef	std err	t	P> t	[0.025	0.975]
const	1.8564	0.4939	3.7587	4.804e-04	0.8622	2.8506
lrm.L1	0.6916	0.0804	8.6006	3.912e-11	0.5297	0.8534
lry.L0	0.5811	0.1442	4.0295	2.077e-04	0.2908	0.8713
lry.L1	-0.2577	0.1592	-1.6185	1.124e-01	-0.5781	0.0628
ibo.L0	-1.1150	0.3531	-3.1576	2.808e-03	-1.8257	-0.4042
ibo.L1	-0.1436	0.4518	-0.3178	7.521e-01	-1.0530	0.7658
ide.L0	0.2716	0.6191	0.4387	6.630e-01	-0.9746	1.5177
ide.L1	0.2275	0.5714	0.3981	6.924e-01	-0.9226	1.3776

On peut fixer un retard différent pour chaque variable exogène en utilisant un dictionnaire python dont les clés sont les noms des variables exogènes et les valeurs sont les retards associés à chacune d'elle.

```
# Retard fixé
exog_lags = {"lry": 1, "ibo": 3, "ide": 2}
ardl2 = ARDL(endog=lrml, lags=[1, 3], exog=exog, order=exog_lags).fit()
ardl2_params = extractParams(ardl2, "lm")
```

Table 6.2 – Coefficients du modèle ARDL(3,1,3,2) creux

	coef	std err	t	P> t	[0.025	0.975]
const	3.1146	0.6511	4.7834	2.355e-05	1.7986	4.4306
lrml.L1	0.4937	0.1522	3.2433	2.387e-03	0.1861	0.8014
lrml.L3	-0.0017	0.0912	-0.0185	9.853e-01	-0.1860	0.1827
lry.L0	0.5869	0.1520	3.8608	4.034e-04	0.2797	0.8941
lry.L1	-0.0677	0.1638	-0.4134	6.815e-01	-0.3987	0.2633
ibo.L0	-1.2426	0.3748	-3.3151	1.954e-03	-2.0002	-0.4851
ibo.L1	0.4016	0.6719	0.5977	5.534e-01	-0.9565	1.7597
ibo.L2	-0.4556	0.6323	-0.7206	4.754e-01	-1.7335	0.8223
ibo.L3	-0.9224	0.4606	-2.0026	5.203e-02	-1.8532	0.0085
ide.L0	0.0111	0.6504	0.0171	9.864e-01	-1.3034	1.3257
ide.L1	-0.0354	0.8430	-0.0419	9.668e-01	-1.7391	1.6684
ide.L2	1.3469	0.6482	2.0779	4.418e-02	0.0368	2.6569

6.1.3 Décalage optimal

Lorsque le vecteur $h = (p, q_1, \dots, q_k)$ du nombre de retards est inconnu, il existe plusieurs façons (non exclusives) permettant de le déterminer : test de significativité des paramètres $\alpha_q, \beta_{1,q_1}, \dots, \beta_{k,q_k}$; critères d'information ou encore tests d'autocorrélation des résidus du type Breusch - Godfrey LM ou Box - Ljung. On présente ici l'approche par les critères d'information.

6.1.3.1 Critère de Akaike (AIC)

Cette technique consiste à retenir comme vecteur h celui qui minimise la fonction de Akaike donnée par :

$$AIC(h) = \log |\widehat{\Sigma}_\varepsilon| + \frac{2}{T} K^2 h \quad (6.6)$$

avec $\widehat{\Sigma}_\varepsilon$, la matrice des variance - covariances des résidus ; T , le nombre d'observations disponible (chaque retard entraîne la perte d'une observation) et K le nombre de régresseurs.

6.1.3.2 Critère de Akaike corrigé (AICC)

Lorsque le nombre de régresseurs K est grand par rapport a nombre d'observations T , c'est - à -dire si $T/K < 40$, il est recommandé d'utiliser le critère d'information d'Akaike corrigé (AICC)¹. Il est défini par :

$$AICC(h) = AIC(h) + \frac{2K(K+1)}{T-K-1} \quad (6.7)$$

1. Voir Hurvich, C. M., & Tsai, C. L. (1995). Model selection for extended quasi-likelihood models in small samples. Biometrics, 1077-1084.

6.1.3.3 Critère de Schwarz (SIC ou BIC)

Cette méthode est proche de celle de Akaike et consiste à retenir le vecteur h qui minimise la fonction de Schwarz :

$$SIC(h) = \log |\hat{\Sigma}_\varepsilon| + \frac{\log(T)}{T} K^2 h \quad (6.8)$$

Le critère BIC est plus parcimonieux que le critère AIC puisqu'il pénalise plus le nombre de variables présentes dans le modèle.

6.1.3.4 Critère de Hannan et Quinn (HQ)

Cette méthode est similaire aux deux techniques précédentes et consiste à retenir comme vecteur, celui qui minimise la fonction de Hannan et Quinn donnée par :

$$HQ(h) = \log |\hat{\Sigma}_\varepsilon| + \frac{2 \log T}{T} K^2 h \quad (6.9)$$

Exemple 6.2 Détermination du décalage optimal

Sous Python, il existe une fonction `ardl_select_order()` du package Statsmodels susceptible de déterminer le décalage optimal d'un modèle ARDL.

```
# Retard optimal
from statsmodels.tsa.ardl import ardl_select_order
res = ardl_select_order(endog=lrn,maxlag=int(np.log(len(data))),exog=exog,
                        maxorder=int(np.log(len(data))),ic='aic')

# Modèle ARDL optimal
ardl3 = ARDL(lrn,res.ar_lags,exog,order=res.dl_lags).fit()
ardl3_params = extractParams(ardl3, "lm")
```

Table 6.3 – Coefficients du modèle ARDL(3,1,3,2)

	coef	std err	t	P> t	[0.025	0.975]
const	2.6202	0.5678	4.6149	4.187e-05	1.4718	3.7686
lrn.L1	0.3192	0.1367	2.3358	2.473e-02	0.0428	0.5956
lrn.L2	0.5326	0.1324	4.0239	2.550e-04	0.2649	0.8003
lrn.L3	-0.2687	0.1021	-2.6305	1.214e-02	-0.4753	-0.0621
lry.L0	0.6728	0.1312	5.1295	8.317e-06	0.4075	0.9381
lry.L1	-0.2574	0.1472	-1.7491	8.815e-02	-0.5551	0.0403
ibo.L0	-1.0785	0.3217	-3.3525	1.790e-03	-1.7292	-0.4278
ibo.L1	-0.1062	0.5858	-0.1813	8.571e-01	-1.2911	1.0787
ibo.L2	0.2877	0.5691	0.5055	6.161e-01	-0.8634	1.4388
ibo.L3	-0.9947	0.3925	-2.5341	1.540e-02	-1.7886	-0.2007
ide.L0	0.1255	0.5545	0.2263	8.222e-01	-0.9960	1.2469
ide.L1	-0.3280	0.7213	-0.4547	6.518e-01	-1.7870	1.1310
ide.L2	1.4079	0.5520	2.5503	1.480e-02	0.2913	2.5245

6.1.4 Prédiction avec un modèle ARDL

On considère le modèle $ARDL(p, q_1, \dots, q_k)$ tel que :

$$Y_t = c + \sum_{i=1}^{i=p} \alpha_i Y_{t-i} + \sum_{j=1}^k \sum_{l=1}^{l=q_j} \beta_{j,l} X_{j,t-l} + \varepsilon_t \quad (6.10)$$

Pour simplifier, on pose

$$Z_t = c + \sum_{j=1}^k \sum_{l=0}^{l=q_j} \beta_{j,l} X_{j,t-l} \quad (6.11)$$

Le modèle ARDL se réécrit sous la forme d'un modèle équivalent à un AR(p)

$$Y_t = Z_t + \alpha_1 Y_{t-1} + \dots + \alpha_p Y_{t-p} + \varepsilon_t \quad (6.12)$$

Conditionnellement à l'ensemble d'information disponible jusqu'à la date T , noté I_T , et à une prévision sur les variables explicatives $\widehat{X}_{j,T}(1) (j = 1, \dots, k)$, la prévision à l'horizon $h = 1$ de la variable dépendante est définie par :

$$\widehat{Y}_T(1) = \widehat{Z}_T(1) + \alpha_1 Y_T + \dots + \alpha_p Y_{T-p+1} \quad (6.13)$$

$$\widehat{Z}_T(1) = c + \sum_{j=1}^k \beta_{j,0} \widehat{X}_{j,T}(1) + \sum_{j=1}^k \sum_{l=0}^{l=q_j-1} \beta_{j,l} X_{j,T-l} \quad (6.14)$$

Dans la pratique, l'erreur de prévision $Y_{T+1} - \widehat{Y}_T(1)$ provient de 3 sources :

1. Les paramètres $c, \alpha_i, \beta_{j,l}$ ne sont pas connus et doivent être estimés, ce qui engendre une erreur d'estimation.
2. Les valeurs futures des k variables explicatives $X_{j,T+1} (j = 1, \dots, k)$ ne sont pas connues. Elles doivent être prévues, ce qui induit une erreur de prévision $X_{j,T+1} - \widehat{X}_{j,T}(1)$ qui se répercute sur l'erreur de prévision sur Y_{T+1} .
3. Par définition, la composante d'erreur de type bruit blanc ε_{T+1} , ne peut pas être prévue puisque $\mathbb{E}(\varepsilon_{T+1} | I_T) = 0$.

En général, la seconde source d'incertitude est négligée car on ne connaît pas la forme ou les propriétés de l'erreur de prévision sur $X_{j,T+1} (j = 1, \dots, k)$. La variance asymptotique de la prévision $\widehat{Y}_T(1)$ et donc l'erreur de prévision sur Y_{T+1} dépend de façon classique de la matrice de variance covariance des paramètres estimés et de la variance du terme d'erreur ε_t .

Le même raisonnement peut être mené pour n'importe quel horizon $h \geq 1$. Par exemple, pour un horizon $h = 2$, la prévision dynamique de Y_{T+2} conditionnellement à l'information I_T disponible à la date T devient :

$$\widehat{Y}_T(2) = \widehat{Z}_T(2) + \alpha_2 \widehat{Y}_T(1) + \alpha_2 Y_T + \dots + \alpha_p Y_{T-p+2} \quad (6.15)$$

$$\widehat{Z}_T(2) = c + \sum_{j=1}^k \sum_{l=0}^{l=1} \beta_{j,l} \widehat{X}_{j,T}(2-l) + \sum_{j=1}^k \sum_{l=2}^{l=q_j-2} \beta_{j,l} X_{j,T-l} \quad (6.16)$$

Dans ce cas, la prévision $\widehat{Y}_T(2)$ nécessite de connaître les prévisions des variables X_1, \dots, X_k aux horizons $h = 1$ et $h = 2$, notées $\widehat{X}_{j,T}(1) (j = 1, \dots, k)$ et $\widehat{X}_{j,T}(2) (j = 1, \dots, k)$. La procédure

de prévisions de Python pour les modèles ARDL, `res.predict()`, nécessite donc de donner les prévisions $\hat{X}_{j,T}(h)$ avec $j = 1, \dots, k$ et $h \geq 1$. L'utilisateur doit donc construire des modèles auxiliaires pour mener à bien ces prévisions ou faire des scénarios sur ces valeurs futures.

La modélisation d'un ensemble de variables et la recherche d'une relation existante entre elles font ressortir deux situations suivant la stationnarité ou non des séries :

1. Si toutes les séries sont stationnaires d'ordre 0, c'est - à - dire stationnaires en niveau, dans ce cas on peut modéliser les données en niveau, en appliquant la méthode des moindres carrés ordinaires ;
2. Si toutes les séries sont stationnaires dans le même ordre (en général d'ordre 1), deux cas de figures se présentent :
 - Si elles ne sont pas cointégrées, on modélise les données en différenciant chaque série et on estime le modèle à l'aide des MCO ;
 - Si elles sont cointégrées, on peut estimer deux modèles : le premier, estimé par la méthode des MCO en utilisant les données en niveau, qui fournira la relation d'équilibre de long terme entre les variables ; et le second, celui de correction d'erreur estimé par la méthode des MCO, qui représentera la dynamique à court terme de la relation entre les variables.

Toutefois, ils arrivent que les séries soient intégrées à des ordres différents et qu'il existe également une possibilité de cointégration entre certaines variables. Alors, les cas présentés ci-dessus ne sont pas valables car les régressions deviennent fallacieuses. D'où l'utilisation du test de cointégration de Pesaran et *al.* (2001).

6.2 Test de cointégration de Pesaran et *al.*(2001)

6.2.1 Rappel sur la cointégration

Dans le chapitre précédent, on a vu que la cointégration entre séries suppose l'existence d'une ou plusieurs relations d'équilibre à long terme entre elles, lesquelles relations pouvant être combinées avec les dynamiques de court terme de ces séries dans un modèle vectoriel à correction d'erreur qui prend la forme :

$$\Delta X_t = \Pi X_{t-1} + \sum_{i=1}^{i=p} \Theta_i \Delta X_{t-i} + \varepsilon_t \quad (6.17)$$

avec ΔX_t le vecteur de variables stationnaires sous études (dont on explique la dynamique) ; Θ_i la matrice dont les éléments sont des paramètres associés à ΔX_{t-i} . Π est une matrice de même dimension que ΔX_t avec $\text{rank}(\Pi)$ le nombre de relations de cointégration.

Pour tester l'existence ou non de la cointégration entre séries, la littérature économétrique fournit plusieurs tests dont l'approche en deux étapes de Engle et Granger et celui de Johansen. Le test de cointégration de Engle et Granger n'aide à vérifier la cointégration qu'entre deux séries intégrées d'ordre 1, il est donc adapté au cas bivarié et s'avère ainsi moins efficace pour des cas multivariés. Le test de cointégration de Johansen permet de vérifier plutôt la cointégration sur plus de deux séries (cas multivariés). Toutefois, bien que le test de Johansen constitue un remède aux limites du test de Engle et Granger pour le cas multivarié, il exige que toutes les séries soient intégrées du même ordre, ce qui n'est pas toujours le cas en pratique.

Cependant, en pratique, les séries ne sont pas toujours intégrées du même ordre, dans ce cas l'on peut recourir au test de cointégration de Pesaran, Shin, and Smith (2001) appelé « *test de*

cointégration aux bornes ». Le modèle ARDL permet d'estimer les dynamiques de court terme et les effets de long terme des séries cointégrées ou même intégrées à des ordres différents.

6.2.2 Le modèle ARDL Pesaran, Shin and Smith (2001)

Le modèle qui sert de base au test de cointégration par les retards échelonnés est la spécification ARDL cointégrée. Au fait, le modèle ARDL Pesaran, Shin, and Smith (2001) est une re-paramétrisation d'un modèle conditionnel à correction d'erreur tiré d'une représentation vectorielle autorégressive. Lorsqu'on étudie la dynamique entre deux séries y_t et x_t , ce modèle prend la forme suivante :

$$\Delta y_t = c_0 + c_1 t + \lambda_1 y_{t-1} + \lambda_2 x_{t-1} + \sum_{i=1}^{i=p-1} \psi'_i \Delta \mathbf{z}_{t-i} + \omega' \Delta x_t + \mu_t \quad (6.18)$$

avec $\mathbf{z}_t = (y_t, x_t)'$ (cf. Pesaran, Shin, and Smith (2001)).

Cette spécification 6.18 présente le modèle ARDL sous forme d'un modèle à correction d'erreur ou d'un VECM, ce qui suppose l'existence des relations de cointégration entre séries. La relation 6.18 peut s'écrire sous la forme suivante :

$$\Delta y_t = c_0 + c_1 t + \sum_{i=1}^{i=p-1} \psi'_i \Delta \mathbf{z}_{t-i} + \omega' \Delta x_t + \theta ECT_{t-1} + \mu_t \quad (6.19)$$

où « ECT_{t-1} » est le terme de correction d'erreur de la relation d'équilibre à long terme vérifiée et « θ » un paramètre indiquant la vitesse d'ajustement au niveau d'équilibre après choc. θ doit être compris entre -1 et 0 et être significatif pour assurer la convergence de la dynamique vers l'équilibre à long terme. -1 signifie une convergence parfaite et instantanée tandis que 0 signifie aucune convergence après un choc dans le processus.

Deux étapes sont à suivre pour appliquer le test de cointégration de Pesaran, à savoir :

- **Etape 1** : Détermination du décalage optimal en utilisant les critères d'informations (AIC, BIC, etc.)
- **Etape 2** : Recours au test de Fisher pour valider les hypothèses suivantes :

$$\begin{cases} H_0 : \lambda_1 = \lambda_2 = 0 : & \text{Existence d'une relation de cointégration} \\ H_1 : \lambda_1 \neq \lambda_2 \neq 0 : & \text{Absence d'une relation de cointégration} \end{cases} \quad (6.20)$$

La procédure du test est telle que l'on devra comparer les valeurs de Fisher obtenues aux valeurs critiques (bornes) simulées pour plusieurs cas et différents seuils par Pesaran, Shin, and Smith (2001). L'on notera des valeurs critiques que la borne supérieure reprend les valeurs critiques pour lesquelles les variables sont intégrées d'ordre 1 et la borne inférieure les valeurs critiques pour lesquelles les variables sont $I(0)$. Ainsi, on a la règle de décision suivante :

- Si $F >$ borne supérieure : Cointégration existe
- Si $F <$ borne inférieure : Cointégration n'existe pas
- Si borne inférieure $< F <$ borne supérieure : Pas de conclusion.

Exemple 6.3 Application du test aux bornes

Nous illustrons le processus de modélisation ARDL en utilisant les données provenant de Wright (2018) sur les préoccupations du public concernant les inégalités aux Etats - Unis. Ces données sont disponibles dans le package `dynamac` de R sous le nom `ineq`.

```
# Chargement des données
ineq_data = r.ineq[["year", "concern", "incshare10", "urate"]].set_index('year')
# Représentation graphique
fig, axe = plt.subplots(1,3,figsize=(16,6))
for idx, name in enumerate(ineq_data.columns):
    axe[idx].plot(ineq_data[name],color = "black");
    axe[idx].set_ylabel(f"{name}");
    axe[idx].set_xlabel("année");
plt.tight_layout();
plt.show()
```

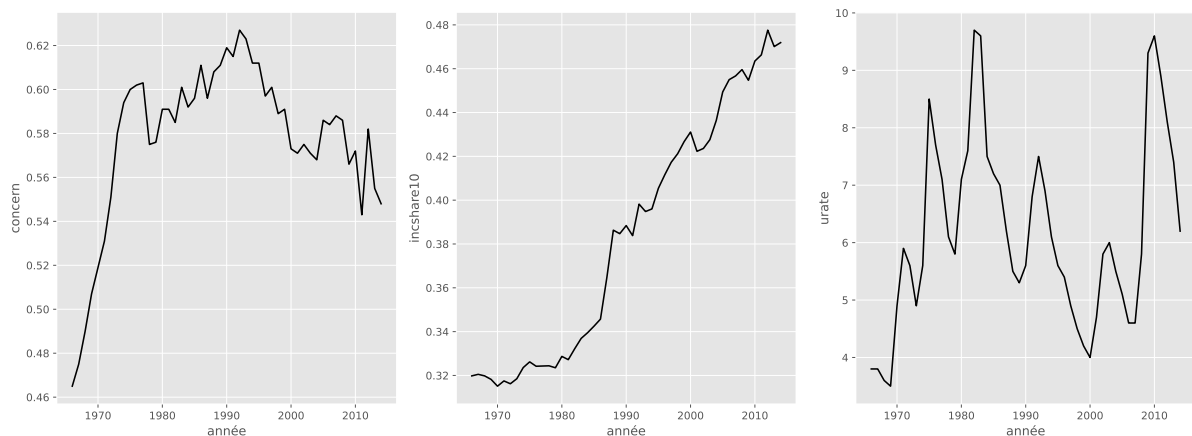


Figure 6.1 – Représentation graphique de concern, incshare10 et urate

Pour notre exemple supposons que les préoccupations du public concernant les inégalités aux Etats - Unis, `concern`, est fonction de la part du revenu allant aux dix pour cent les plus riches, Income Top 10 (`incshare10`). Nous émettons également l'hypothèse que le taux de chômage, `urate`, affecte les préoccupations à court terme (c'est - à - dire qu'il ne cointègre pas).

Etape 1 : Test de stationnarité

Avant d'effectuer le test aux bornes de Pesaran, nous devons d'abord vérifier la stationnarité des séries. Nous appliquons le test ADF

— Test ADF sur `concern`

```
# Augmented Dickey - Fuller : Modèle [3] sur concern
from arch.unitroot import ADF
regconcern = ADF(ineq_data.concern,lags=1,trend = 'ct')
regconcern_res = extractUnitRootParams(regconcern)
```

La statistique est supérieure à la valeur critique au seuil de 5%. Le processus `concern` est non stationnaire en niveau. On passe au test sur la différence première.

— Test ADF sur $\Delta\text{concern}$

Table 6.4 – Modèle [3] du test ADF sur concern

	stat	p-value	1%	5%	10%
test	-2.889271	0.1660873	-4.165551	-3.50843	-3.184055

```
# Augmented Dickey - Fuller : Modèle [3] sur \Deltaconcern
regconcern2 = ADF(ineq_data.concern.diff().dropna(),lags=1,trend = 'ct')
regconcern2_res = extractUnitRootParams(regconcern2)
```

Table 6.5 – Modèle [3] du test ADF sur Δ concern

	stat	p-value	1%	5%	10%
test	-4.404055	0.0021573	-4.17039	-3.51067	-3.185344

La statistique est inférieure à la valeur critique au seuil de 5%. Le processus `concern` est stationnaire en différence première.

— Test ADF sur `incshare10`

```
# Augmented Dickey - Fuller : Modèle [3] sur incshare10
regincshare = ADF(ineq_data.incshare10,lags=1,trend = 'ct')
regincshare_res = extractUnitRootParams(regincshare)
```

Table 6.6 – Modèle [3] du test ADF sur `incshare10`

	stat	p-value	1%	5%	10%
test	-2.854099	0.1777843	-4.165551	-3.50843	-3.184055

La statistique est supérieure à la valeur critique au seuil de 5%. Le processus `incshare10` est non stationnaire en niveau. On passe au test sur la différence première.

— Test ADF sur Δ `incshare10`

```
# Augmented Dickey - Fuller : Modèle [3] sur \Deltaincshare10
regincshare2 = ADF(ineq_data.incshare10.diff().dropna(),lags=1,trend = 'ct')
regincshare2_res = extractUnitRootParams(regincshare2)
```

Table 6.7 – Modèle [3] du test ADF sur Δ `incshare10`

	stat	p-value	1%	5%	10%
test	-4.228043	0.0040594	-4.17039	-3.51067	-3.185344

La statistique est inférieure à la valeur critique au seuil de 5%. Le processus `incshare10` est stationnaire en différence première.

— Test ADF sur `urate`

```
# Augmented Dickey - Fuller : Modèle [3] sur urate
from arch.unitroot import ADF
regurate = ADF(ineq_data.urate,lags=1,trend = 'ct')
regurate_res = extractUnitRootParams(regurate)
```

Table 6.8 – Modèle [3] du test ADF sur *urate*

	stat	p-value	1%	5%	10%
test	-3.559453	0.0334468	-4.165551	-3.50843	-3.184055

La statistique est supérieure à la valeur critique au seuil de 5%. Le processus *urate* est non stationnaire en niveau. On passe au test sur la différence première.

— Test ADF sur Δurate

```
# Augmented Dickey - Fuller : Modèle [3] sur \Delta urate
regurate2 = ADF(ineq_data.urate.diff().dropna(),lags=1,trend = 'ct')
regurate2_res = extractUnitRootParams(regconcern2)
```

Table 6.9 – Modèle [3] du test ADF sur Δurate

	stat	p-value	1%	5%	10%
test	-4.404055	0.0021573	-4.17039	-3.51067	-3.185344

La statistique est inférieure à la valeur critique au seuil de 5%. Le processus *urate* est stationnaire en différence première.

Les trois séries sont intégrées d'ordre $I(1)$, car elles apparaissent intégrées dans les niveaux mais stationnaires dans les différences premières.

Etape 2 : Modèle à correction d'erreur

Etant donné que toutes les séries semblent être $I(1)$, nous procédons à l'estimation d'un modèle sous forme de correction d'erreur, puis nous testons la cointégration entre le souci d'inégalité et la part de revenu des 10% les plus riches. Notre modèle de correction d'erreur apparaît comme :

$$\Delta\text{concern}_t = \alpha_0 + \phi_1\text{concern}_{t-1} + \alpha_1\text{incshare10}_{t-1} + \theta_1\Delta\text{incshare10}_t + \psi_1\Delta\text{urate}_t + \varepsilon_t$$

où l'on suppose $\varepsilon_t \sim \mathcal{N}(0, \sigma_\varepsilon^2)$.

Nous estimons notre modèle ECM en utilisant la fonction `ols` de la librairie `Statsmodels`. Pour cela, nous créons d'abord la fonction `diff()` qui prendra en compte les variables en différence première.

```
#Difference operator:
def diff(x):
    if isinstance(x, pd.Series):
        return x.diff()
    else:
        x = pd.Series(x)
        return x.diff()
# Estimation des paramètres par MCO
import statsmodels.formula.api as smf
fm = " ".join((
    "diff(concern)~lag(concern,1)",
    "+diff(incshare10)+diff(urate)+lag(incshare10,1)"
))
```

```

))
regecm = smf.ols(fm,data=ineq_data).fit()
regecm_params = extractParams(regecm, "lm")

```

Table 6.10 – Coefficients estimés du modèle à correction d'erreur

	coef	std err	t	P> t	[0.025	0.975]
Intercept	0.1220	0.0280	4.3638	7.874e-05	0.0656	0.1784
lag(concern, 1)	-0.1677	0.0487	-3.4425	1.296e-03	-0.2659	-0.0694
diff(incshare10)	0.8006	0.2966	2.6990	9.900e-03	0.2024	1.3988
diff(urate)	0.0011	0.0017	0.6583	5.138e-01	-0.0023	0.0045
lag(incshare10, 1)	-0.0680	0.0318	-2.1370	3.833e-02	-0.1322	-0.0038

Comme le montrent les résultats de la régression qui a inclu une constante, la variable dépendante retardée, `lag(concern, 1)`, la première différence des deux régresseurs (Revenu Top 10 et Chômage), ainsi que le retard de l'Income Top 10 (*cf.* Tableau 6.10), alors les variations du revenu Top 10 affectent les variations de la préoccupation à court terme, les variations du chômage n'ont pas d'effet statistiquement significatif à court terme.

Le décalage du revenu Top 10 est négatif et statistiquement significatif. De plus, le paramètre sur la variable dépendante retardée est négatif, entre 0 et -1 , et statistiquement significatif, ce qui nous donne rapidement preuve d'un processus de cointégration en cours.

```

# AIC et BIC
new_df = pd.DataFrame({
    "Log - likelihood" : regecm.llf, "AIC" : regecm.aic, "BIC" : regecm.bic,
    "k" : len(regecm.params), "T" : len(regecm.resid)
}, index = ["valeur"])

```

Table 6.11 – Statistiques d'ajustement

	Log - likelihood	AIC	BIC	k	T
valeur	148.094	-286.187	-276.831	5	48

Etape 3 : Diagnostics post - estimation

Une composante essentielle de la modélisation ARDL est de s'assurer que les résidus de toute estimation ARDL sont du bruit blanc. L'un des symptômes de l'autocorrélation résiduelle en présence de la variable dépendante retardée est que les MCO entraîneront des estimations biaisées et incohérentes.

— Test de normalité : Shapiro - Wilk

Le test de Shapiro – Wilk teste l'hypothèse nulle selon laquelle un échantillon x_1, \dots, x_n est issu d'une population normalement distribuée. Si la p-value est inférieure à un niveau α choisi (par exemple 0.05), alors l'hypothèse nulle est rejetée.

```

# Test de normalité de Shapiro - Wilk
import scipy.stats as st
shapiro_test = st.shapiro(regecm.resid)
shapiro_res = pd.DataFrame({
    "W statistic" : shapiro_test[0], "p-value" : shapiro_test[1]
}, index = ["Shapiro test"])

```

Table 6.12 – Test de normalité de Shapiro - Wilk

	W statistic	p-value
Shapiro test	0.965	0.158

Au seuil de 5%, la p-value est non significative. On ne rejette pas l'hypothèse nulle : les résidus du modèle suivent une loi normale.

— Test d'autocorrélation des résidus : Breusch - Godfrey

Le test de Breusch - Godfrey est un test statistique qui teste l'autocorrélation de n'importe quel ordre. Il s'agit d'un test asymptotique qui teste directement la significativité du coefficient ρ dans la formule :

$$\varepsilon_t = \rho\varepsilon_{t-1} + \eta_t \quad (6.21)$$

où η_t est un bruit blanc.

L'hypothèse nulle (H_0) stipule qu'il y a non auto-corrélation donc $\rho = 0$. L'hypothèse de recherche (H_1) stipule qu'il y a auto-corrélation donc ρ différent de 0 avec toujours $|\rho| < 1$.

```
# Test d'autocorrélation de Breusch - Godfrey
import statsmodels.stats.diagnostic as dg
def lmtest(model,lag):
    return dg.acorr_breusch_godfrey(model,nlags=lag)
# Application
lm_test = lmtest(regecm,1)
lm_res = pd.DataFrame({
    "LM statistic " : lm_test[0], "LM pvalue" : lm_test[1],
    "F-statistic " : lm_test[2], "F-pvalue" : lm_test[3]
}, index = ["Breusch - Godfrey test"])
```

Table 6.13 – Test de Breusch - Godfrey d'autocorrélation des résidus

	LM statistic	LM pvalue	F-statistic	F-pvalue
Breusch - Godfrey test	3.704	0.054	3.512	0.068

Au seuil de 5%, les p-values sont non significatives, on accepte l'hypothèse nulle d'absence d'autocorrélation des résidus.

— Test d'hétéroscédasticité : ARCH - LM

Le test d'hétéroscédasticité utilisé est le test ARCH - LM.

```
# Test d'hétéroscédasticité : ARCH - LM
from statsmodels.stats.diagnostic import het_arch
hettest = het_arch(regecm.resid,nlags=1)
arch_test = pd.DataFrame({
    "LM statistic" : hettest[0], "LM p-value" : hettest[1],
    "f-statistic" : hettest[2], "f p-value" : hettest[3]
}, index = ["ARCH LM test"])
```

Au seuil de 5%, les p-values sont non significatives, on accepte l'hypothèse nulle d'homoscédasticité des résidus.

Table 6.14 – Test ARCH-LM d'hétéroscédasticité des résidus

	LM statistic	LM p-value	f-statistic	f p-value
ARCH LM test	1.423	0.233	1.405	0.242

— Test de nullité des coefficients

Il consiste à tester l'hypothèse nulle suivante : $H_0 : \phi_1 = \alpha_1 = 0$. Ceci revient à tester $R\beta = 0$ avec

$$R = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix} \quad \text{et} \quad \beta = (\alpha_0, \phi_1, \alpha_1, \theta_1, \psi_1)'$$

```
# Test de Fisher de nullité des coefficients phi_1 et alpha_1
f_test = regecm.f_test(np.array([[0,1,0,0,0],[0,0,1,0,0]]))
print(f_test)
```

```
## <F test: F=8.26726983527152, p=0.0009163012859149706, df_denom=43, df_num=2>
```

Etape 4 : Test de cointégration aux bornes

Ce test est basé sur le modèle suivant :

$$\Delta concern_t = \alpha_0 + \phi_1 concern_{t-1} + \alpha_1 incshare10_{t-1} + \gamma_1 \Delta concern_{t-1} + \theta_1 \Delta incshare10_t + \psi_1 \Delta urate_t + \varepsilon_t$$

où l'on suppose $\varepsilon_t \sim \mathcal{N}(0, \sigma_\varepsilon^2)$.

On effectue le test de nullité $R\beta = 0$ avec

$$R = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \quad \text{et} \quad \beta = (\alpha_0, \phi_1, \alpha_1, \gamma_1, \theta_1, \psi_1)$$

```
# Modèle UECM
fm2= " ".join((
    "diff(concern)~lag(concern,1)+",
    "lag(incshare10,1)+lag(diff(concern),1)+diff(incshare10)+diff(urate)"
))
regecm2 = smf.ols(fm2,data=ineq_data).fit()
f_test2 = regecm2.f_test(np.array([[0,1,0,0,0,0],[0,0,1,0,0,0]]))
print(f_test2)
```

```
## <F test: F=12.203505710394767, p=6.946576040936252e-05, df_denom=41, df_num=2>
```

La statistique du test vaut $F = 12.204$ et les valeurs critiques du test sont :

```
# Valeurs critiques
f_crit_val = pd.DataFrame({
    "I(0)" : [4.19, 5.22, 7.56], "I(1)" : [4.94, 6.07, 8.685]
},index = ["10% critical value","5% critical value", "1% critical value"])
```

Table 6.15 – Valeurs critiques de F

	I(0)	I(1)
10% critical value	4.19	4.940
5% critical value	5.22	6.070
1% critical value	7.56	8.685

Dans notre exemple, la valeur de la statistique F dépasse la valeur critique à la borne supérieure $I(1)$ du test au niveau de 1%, nous pouvons conclure que les revenus et les préoccupations concernant les inégalités sont dans une relation de cointégration.

```
# Test t-student
t_test = regecm2.t_test(np.array([[0,1,0,0,0,0],[0,0,1,0,0,0]]))
print(t_test)

##                                Test for Constraints
## =====
##          coef      std err          t      P>|t|      [0.025      0.975]
## -----
## c0          -0.1951      0.053     -3.684      0.001     -0.302     -0.088
## c1          -0.0858      0.033     -2.615      0.012     -0.152     -0.020
## =====

# Valeurs critiques
t_crit_val = pd.DataFrame({
    "I(0)" : [-2.57,-2.86,-3.43], "I(1)" : [-2.91, -3.22, -3.82]
},index = ["10% critical value","5% critical value", "1% critical value"])
```

Table 6.16 – Valeurs critiques de t

	I(0)	I(1)
10% critical value	-2.57	-2.91
5% critical value	-2.86	-3.22
1% critical value	-3.43	-3.82

Etant donné que la statistique t de $t_{c0} = -3.684$ tombe en dessous du seuil de la valeur critique $I(1)$ de 5%, cela soutient davantage la cointégration. Pris ensemble, ces résultats indiquent qu'il existe une relation de cointégration entre les préoccupations concernant les inégalités et la part des revenus des 10% les plus riches, et que l'équation ECM est correctement spécifiée.

Exemple 6.4 *West German dataset*

Les données avec lesquelles nous allons travailler sont extraites du livre [New Introduction to Multiple Time Series Analysis](http://www.stata-press.com/data/r12/lutkepohl2.dta) de Lütkepohl (2005) disponible <http://www.stata-press.com/data/r12/lutkepohl2.dta>. Nous travaillons avec les données sous forme logarithmique.

```
# Chargement des données
file = "http://www.stata-press.com/data/r12/lutkepohl2.dta"
german = pd.read_stata(file).set_index("qtr")
german.index = pd.date_range(start="1960-03-31",periods=len(german),freq="3M")
# Représentation graphique
```

```
fig, axe = plt.subplots(figsize=(16,5))
german[["ln_inv", "ln_inc", "ln_consump"]].plot(ax=axe);
axe.set_xlabel("année");
plt.show()
```

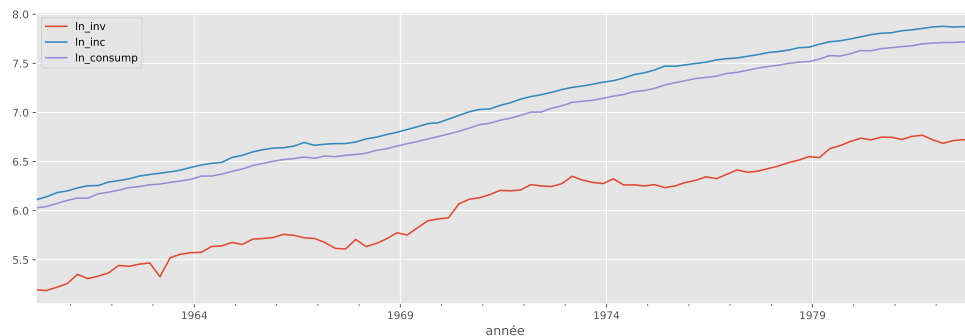


Figure 6.2 – Evolution des séries invest/income/consump sous forme logarithmique

Nous estimons le modèle ARDL à correction d'erreur suivant :

$$\Delta \ln(Investment)_t = \alpha_0 + \theta_0 \ln(Investment)_{t-1} + \theta_1 \ln(Income)_t + \theta_2 \ln(Consumption)_{t-1} + \beta_1 \Delta \ln(Income)_t + \beta_2 \Delta \ln(Consumption)_t + \varepsilon_t$$

Sous Python, la fonction `UECM` du module `Statsmodels` permet d'estimer le modèle à correction d'erreur sous la forme d'un modèle ARDL.

```
# Estimation du modèle ECM
from statsmodels.tsa.api import UECM
reguecm = UECM.from_formula("ln_inv~ln_inc+ln_consump",data=german,lags=1,order=1,
                             trend='c').fit()
reguecm_params = extractParams(reguecm,"lm")
```

Table 6.17 – Coefficients estimés du modèle à correction d'erreur

	coef	std err	t	P> t	[0.025	0.975]
const	-0.0080	0.0711	-0.1126	9.106e-01	-0.1494	0.1334
ln_inv.L1	-0.1398	0.0589	-2.3721	1.994e-02	-0.2570	-0.0226
ln_inc.L1	-0.2095	0.3337	-0.6277	5.319e-01	-0.8730	0.4541
ln_consump.L1	0.3359	0.3333	1.0076	3.165e-01	-0.3269	0.9986
D.ln_inc.L0	-0.2007	0.4589	-0.4374	6.630e-01	-1.1130	0.7117
D.ln_consump.L0	1.5482	0.5375	2.8803	5.026e-03	0.4795	2.6170

On teste l'hypothèse nulle suivante : $H_0 : \theta_0 = \theta_1 = \theta_2 = 0$, c'est - à - dire $R\beta = 0$ avec

$$R = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \quad \text{et} \quad \beta = (\alpha_0, \theta_0, \theta_1, \theta_2, \beta_1, \beta_2)$$

```
# Test de nullité des coefficients
f_test = reguecm.f_test(np.array([[0,1,0,0,0,0],[0,0,1,0,0,0],[0,0,0,1,0,0]]))
print(f_test)
```

```
## <F test: F=2.6016533050153137, p=0.05728826853917501, df_denom=85, df_num=3>
```

— Test de cointégration de Pésaran

Sous Python, la méthode `model.bounds_test` permet d'implémenter le test de cointégration au sens de Pesaran, Shin et Smith (2001).

```
# ARDL bound test
def bound_f_test(model,**kwargs):
    bound_test = model.bounds_test(**kwargs)
    print("Cointegration bounds test of Pesaran, Shin, and Smith (2001)")
    print("H0 : "+ bound_test[3] + " vs H1 : "+ bound_test[4])
    print(f"F-statistic = {round(bound_test[0],3)}"); print("Critical values")
    print(bound_test[1])
# Application
bound_f_test(reguecm,case = 3)
```

```
## Cointegration bounds test of Pesaran, Shin, and Smith (2001)
## H0 : No Cointegration vs H1 : Possible Cointegration
## F-statistic = 2.602
## Critical values
##          lower      upper
## percentile
## 90.0      2.729522  3.747156
## 95.0      3.229022  4.322282
## 99.0      4.311041  5.542552
## 99.9      5.759921  7.143546
```

Un recul doit être toujours pris dans la manipulation des logiciels. Comme on peut le constater, l'application du test de cointégration au sens de Pesaran pour notre modèle ARDL à correction d'erreur donne la même valeur pour la statistique F, mais des valeurs critiques différentes, au seuil de 5%.

```
# Application sous R
ardl.model <- dynardl(ln_inv~ln_inc+ln_consump, data = py$german,
                      lags = list("ln_inv" = 1, "ln_inc" = 1, "ln_consump" = 1),
                      diffs = c("ln_inc", "ln_consump"),ec = TRUE, simulate = FALSE)

## [1] "Error correction (EC) specified; dependent variable to be run in differences."

pssbounds(ardl.model)

##
## PESARAN, SHIN AND SMITH (2001) COINTEGRATION TEST
##
```

```

## Observations: 91
## Number of Lagged Regressors (not including LDV) (k): 2
## Case: 3 (Unrestricted intercept; no trend)
##
## -----
## -                      F-test                      -
## -----
##      <----- I(0) ----- I(1) ----->
## 10% critical value      3.17      4.14
## 5% critical value      3.79      4.85
## 1% critical value      5.15      6.36
##
##
## F-statistic = 2.602
## -----
## -                      t-test                      -
## -----
##      <----- I(0) ----- I(1) ----->
## 10% critical value     -2.57     -3.21
## 5% critical value     -2.86     -3.53
## 1% critical value     -3.43     -4.10
##
##
## t statistic = -2.372
## -----
## F-statistic note: Asymptotic critical values used.
## t-statistic note: Asymptotic critical values used.

```

Pour notre modèle, la statistique $F = 2.60$ est inférieure aux trois bornes inférieures. Par conséquent, il y a absence de cointégration au sens de Pesaran pour nos variables.

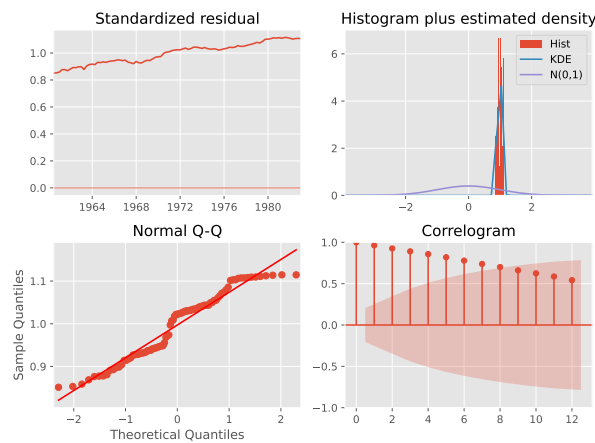
Diagnostics du modèle

La méthode `res.plot_diagnostics()` retourne ensemble de graphique sur les résidus standardisés : QQ-plot, ACF, etc.

```

# Graphique des résidus standardisés
reguecm.plot_diagnostics(lags=12, figsize=(8,6))
plt.tight_layout();
plt.show()

```



— Test de normalité

La méthode `model.test_normality()` teste l'adéquation des résidus à la loi normale. Cette méthode utilise le test de Jarque - Bera.

```
# Test de normalité : Jarque - Bera
jb_norm = regucom.test_normality()
jb_test = pd.DataFrame({
    "statistic" : jb_norm[0], "p-value" : jb_norm[1],
    "Skweness" : jb_norm[2], "Kurtosis" : jb_norm[3]
}, index = ["Jarque - Bera test"])
```

Table 6.18 — Test de normalité de Jarque - Bera des résidus

	statistic	p-value	Skweness	Kurtosis
Jarque - Bera test	5.4511	0.0655	-0.1213	1.8258

— Test d'hétéroscédasticité

La méthode `model.test_heteroskedasticity()` effectue le test ARCH-LM d'hétéroscédasticité. Elle prend en entrée un paramètre `lags` afin de spécifier le retard maximal du test ARCH-LM.

```
# Test d'hétéroscédasticité : ARCH - LM
arch_test = regucom.test_heteroskedasticity(lags=10)
print(arch_test)
```

```
##          ARCH-LM          P-value  DF
## Lag
## 1      89.135790  3.686214e-21    1
## 2      88.119370  7.330304e-20    2
## 3      87.111407  9.138570e-19    3
## 4      86.163608  8.591513e-18    4
## 5      85.227706  6.744103e-17    5
## 6      84.234996  4.754597e-16    6
## 7      83.217754  3.034730e-15    7
## 8      82.207077  1.755943e-14    8
## 9      81.200333  9.330036e-14    9
## 10     80.220757  4.544319e-13   10
```

La méthode `res.diagnostic_summary()` retourne une série de test de diagnostic du modèle.

```
# Diagnostic du modèle
print(reguecm.diagnostic_summary())
```

```
##                      Test of Normality
## =====
## Jarque-Bera      P-value    Skewness    Kurtosis
## -----
##          5.451      0.066      -0.121      1.826
##
##
##      Test of Conditional Homoskedasticity
## =====
##          Lag    ARCH-LM    P-value    DF
## -----
##           1      89.136      0.000      1
##           2      88.119      0.000      2
##           3      87.111      0.000      3
##           4      86.164      0.000      4
##           5      85.228      0.000      5
##           6      84.235      0.000      6
##           7      83.218      0.000      7
##           8      82.207      0.000      8
##           9      81.200      0.000      9
##          10      80.221      0.000     10
## -----
```

Tables statistiques

Table 19 – Distribution cumulative empirique de t_ρ sous $H_0 : \rho = 0$

	Probabilité d'une valeur inférieure							
T	0.01	0.025	0.05	0.10	0.90	0.95	0.975	0.99
Modèle [1] : t_{ρ^\bullet}								
25	-2.66	-2.26	-1.95	-1.60	0.92	1.33	1.70	2.16
50	-2.62	-2.25	-1.95	-1.61	0.91	1.31	1.66	2.08
100	-2.60	-2.24	-1.95	-1.61	0.90	1.29	1.64	2.03
250	-2.58	-2.23	-1.95	-1.62	0.89	1.29	1.63	2.01
500	-2.58	-2.23	-1.95	-1.62	0.89	1.28	1.62	2.00
∞	-2.58	-2.23	-1.95	-1.62	0.89	1.28	1.62	2.00
Modèle [2] : $t_{\hat{\rho}}$								
25	-3.75	-3.33	-3.00	-2.63	-0.37	0.00	0.34	0.72
50	-3.58	-3.22	-2.93	-2.60	-0.40	-0.03	0.29	0.66
100	-3.51	-3.17	-2.89	-2.58	-0.42	-0.05	0.26	0.63
250	-3.46	-3.14	-2.88	-2.57	-0.42	-0.06	0.24	0.62
500	-3.44	-3.13	-2.87	-2.57	-0.43	-0.07	0.24	0.61
∞	-3.43	-3.12	-2.89	-2.57	-0.44	-0.07	0.23	0.60
Modèle [3] : $t_{\tilde{\rho}}$								
25	-4.38	-3.95	-3.60	-3.24	-1.14	-0.80	-0.50	-0.15
50	-4.15	-3.80	-3.50	-3.18	-1.19	-0.87	-0.58	-0.24
100	-4.04	-3.73	-3.45	-3.15	-1.22	-0.90	-0.62	-0.28
250	-3.99	-3.69	-3.43	-3.13	-1.23	-0.92	-0.64	-0.31
500	-3.98	-3.68	-3.42	-3.13	-1.24	-0.93	-0.65	-0.32
∞	-3.96	-3.66	-3.41	-3.12	-1.25	-0.94	-0.66	-0.33

Source : Fuller 1976, p. 373, Tableau 8.5.2.

Table 20 – Distribution empirique de t pour $(c, \rho) = (0, 0)$ Modèle [2]

	Probabilité d'une valeur inférieure			
T	0.90	0.95	0.975	0.99
Modèle [2] : $t_{\hat{c}}$				
25	2.20	2.61	2.97	3.41
50	2.18	2.56	2.89	3.28
100	2.17	2.54	2.86	3.22
250	2.16	2.53	2.84	3.19
500	2.16	2.52	2.83	3.18
∞	2.16	2.52	2.83	3.18

Source : Dickey et Fuller, 1981, p1062, Tableau I.

Table 21 – Distribution empirique de Φ_1 pour $H_0^1 : (c, \rho) = (0, 0)$
Modèle [2]

	Probabilité d'une valeur inférieure							
T	0.01	0.025	0.05	0.10	0.90	0.95	0.975	0.99
Modèle [2] $\Phi_1 H_0^1 : (c, \rho) = (0, 0)$								
25	0.29	0.38	0.49	0.65	4.12	5.18	6.30	7.88
50	0.29	0.39	0.50	0.66	3.94	4.86	5.80	7.06
100	0.29	0.39	0.50	0.67	3.86	4.71	5.57	6.70
250	0.30	0.39	0.51	0.67	3.81	4.63	5.45	6.52
500	0.30	0.39	0.51	0.67	3.79	4.61	5.41	6.47
∞	0.30	0.40	0.51	0.67	3.78	4.59	5.38	6.43

Source : Dickey et Fuller, 1981, p1063, Tableau IV.

Table 22 – Distribution empirique de $t_{\hat{c}}$
pour $(c, b, \rho) = (0, 0, 0)$ Modèle [3]

	Probabilité d'une valeur inférieure			
T	0.90	0.95	0.975	0.99
Modèle [3] : $t_{\hat{c}}$				
25	2.77	3.20	3.59	4.05
50	2.75	3.14	3.47	3.87
100	2.73	3.11	3.42	3.78
250	2.73	3.09	3.39	3.74
500	2.72	3.08	3.38	3.72
∞	2.72	3.08	3.38	3.71

Source : Dickey et Fuller, 1981, p1062, Tableau II.

Table 23 – Distribution empirique de $t_{\hat{b}}$
pour $(c, b, \rho) = (0, 0, 0)$ Modèle [3]

	Probabilité d'une valeur inférieure			
T	0.90	0.95	0.975	0.99
Modèle [3] : $t_{\hat{b}}$				
25	2.39	2.85	3.25	3.74
50	2.38	2.81	3.18	3.60
100	2.38	2.79	3.14	3.53
250	2.38	2.79	3.12	3.49
500	2.38	2.78	3.11	3.48
∞	2.38	2.78	3.11	3.46

Source : Dickey et Fuller, 1981, p1062, Tableau III.

Table 24 – Distribution empirique de Φ_2 pour $H_0^2 : (c, b, \rho) = (0, 0, 0)$
Modèle [3]

	Probabilité d'une valeur inférieure							
T	0.01	0.025	0.05	0.10	0.90	0.95	0.975	0.99
Modèle [3] $\Phi_2 H_0^2 : (c, b, \rho) = (0, 0, 0)$								
25	0.61	0.75	0.89	1.10	4.67	5.68	6.75	8.21
50	0.62	0.77	0.91	1.12	4.31	5.13	5.94	7.02
100	0.63	0.77	0.92	1.12	4.16	4.88	5.59	6.50
250	0.63	0.77	0.92	1.13	4.07	4.75	5.40	6.22
500	0.63	0.77	0.92	1.13	4.05	4.71	5.35	6.15
∞	0.63	0.77	0.92	1.13	4.03	4.68	5.31	6.09

Source : Dickey et Fuller, 1981, p1063, Tableau V.

Table 25 – Distribution empirique de Φ_3 pour $H_0^3 : (c, b, \rho) = (c, 0, 0)$
Modèle [3]

	Probabilité d'une valeur inférieure							
T	0.01	0.025	0.05	0.10	0.90	0.95	0.975	0.99
Modèle [3] $\Phi_3 H_0^3 : (c, b, \rho) = (c, 0, 0)$								
25	0.74	0.90	1.08	1.33	5.91	7.24	8.65	10.61
50	0.76	0.93	1.11	1.37	5.61	6.73	7.81	9.31
100	0.76	0.94	1.12	1.38	5.47	6.49	7.44	8.73
250	0.76	0.94	1.13	1.39	5.39	6.34	7.25	8.43
500	0.76	0.94	1.13	1.39	5.36	6.30	7.20	8.34
∞	0.77	0.94	1.13	1.39	5.34	6.25	7.16	8.27

Source : Dickey et Fuller, 1981, p1063, Tableau VI.

Bibliographie

- Abraham, Bovas, and Johannes Ledolter. 1983. *Statistical Methods for Forecasting*. Vol. 179. Wiley Online Library.
- Akaike, Hirotugu. 1969. "Fitting Autoregressive Models for Prediction." *Annals of the Institute of Statistical Mathematics* 21 (1) : 243–47.
- . 1973. "Information Theory and an Extension of the Maximum Likelihood Principle." In *2nd International Symposium on Information Theory, 1973*, 267–81. Akademiai Kiado.
- . 1974. "A New Look at the Statistical Model Identification." *IEEE Transactions on Automatic Control* 19 (6) : 716–23.
- Anderson, TW. 1984. "An Introduction to Multivariate Analysis Techniques." New York : Wiley.
- Andrews, Donald WK. 1987. "Asymptotic Results for Generalized Wald Tests." *Econometric Theory* 3 (3) : 348–58.
- Aragon, Yves. 2011. *Séries Temporelles*. Ed. Techniques Ingénieur.
- Avram, Florin. 2017. "Séries Temporelles Avec r." Edition.
- Basu, AK, and S Sen Roy. 1987. "On Asymptotic Prediction Problems for Multivariate Autoregressive Models in the Unstable Nonexplosive Case." *Calcutta Statistical Association Bulletin* 36 (1-2) : 29–38.
- Bourbonnais, Régis. 2018. *Économétrie-10e éd. : Cours Et Exercices Corrigés*. Dunod.
- Bourbonnais, Régis, Michel Terraza, et al. 2022. "Analyse Des séries Temporelles."
- Box, George EP, and David A Pierce. 1970. "Distribution of Residual Autocorrelations in Autoregressive-Integrated Moving Average Time Series Models." *Journal of the American Statistical Association* 65 (332) : 1509–26.
- Breusch, Trevor S. 1978. "Testing for Autocorrelation in Dynamic Linear Models." *Australian Economic Papers* 17 (31) : 334–55.
- Brockwell, Peter J, and Richard A Davis. 2002. *Introduction to Time Series and Forecasting*. Springer.
- . 2009. *Time Series : Theory and Methods*. Springer science & business media.
- Brüggemann, Ralf, Helmut Lütkepohl, and Pentti Saikkonen. 2006. "Residual Autocorrelation Testing for Vector Error Correction Models." *Journal of Econometrics* 134 (2) : 579–604.
- Charpentier, Arthur. 2006. "Cours de séries Temporelles : Théorie Et Applications." *Université Paris Dauphine*.
- . 2012. "Modèles de Prévision séries Temporelles." *Paris, France*.
- Davies, Nelville, CM Triggs, and Paul Newbold. 1977. "Significance Levels of the Box-Pierce Portmanteau Statistic in Finite Samples." *Biometrika* 64 (3) : 517–22.
- Dolado, Juan J, and Helmut Lütkepohl. 1996. "Making Wald Tests Work for Cointegrated VAR Systems." *Econometric Reviews* 15 (4) : 369–86.

- Edgerton, David, and Ghazi Shukur. 1999. "Testing Autocorrelation in a System Perspective Testing Autocorrelation." *Econometric Reviews* 18 (4) : 343–86.
- Engle, Robert F, and Clive WJ Granger. 1987. "Co-Integration and Error Correction : Representation, Estimation, and Testing." *Econometrica : Journal of the Econometric Society*, 251–76.
- Engle, Robert F, David F Hendry, and Jean-Francois Richard. 1983. "Exogeneity." *Econometrica : Journal of the Econometric Society*, 277–304.
- Godfrey, Leslie G. 1978. "Testing for Higher Order Serial Correlation in Regression Equations When the Regressors Include Lagged Dependent Variables." *Econometrica : Journal of the Econometric Society*, 1303–10.
- Godfrey, LG. 1988. "Misspecification Tests in Econometrics. Cambridge, UK : Cambridge University Press."
- Granger, Clive WJ. 1969. "Investigating Causal Relations by Econometric Models and Cross-Spectral Methods." *Econometrica : Journal of the Econometric Society*, 424–38.
- . 1981. "Some Properties of Time Series Data and Their Use in Econometric Model Specification." *Journal of Econometrics* 16 (1) : 121–30.
- Granger, Clive WJ, and Paul Newbold. 1974. "Spurious Regressions in Econometrics." *Journal of Econometrics* 2 (2) : 111–20.
- Granger, Clive WJ, and Andrew A Weiss. 1983. "Time Series Analysis of Error-Correction Models." In *Studies in Econometrics, Time Series, and Multivariate Statistics*, 255–78. Elsevier.
- Griffiths, William E, George G Judge, R Carter Hill, Helmut Lütkepohl, and Tsoung-Chao Lee. 1985. *The Theory and Practice of Econometrics*. Wiley.
- Hallin, Marc, and Guy Mélard. 1988. "Rank-Based Tests for Randomness Against First-Order Serial Dependence." *Journal of the American Statistical Association* 83 (404) : 1117–28.
- Hannan, Edward J, and Barry G Quinn. 1979. "The Determination of the Order of an Autoregression." *Journal of the Royal Statistical Society : Series B (Methodological)* 41 (2) : 190–95.
- Haslwanter, Thomas. 2016. "An Introduction to Statistics with Python." *With Applications in the Life Sciences. Switzerland : Springer International Publishing*.
- Hendry, David F, Adrian R Pagan, and J Denis Sargan. 1984. "Dynamic Specification." *Handbook of Econometrics* 2 : 1023–1100.
- Hosking, Jonathan RM. 1980. "The Multivariate Portmanteau Statistic." *Journal of the American Statistical Association* 75 (371) : 602–8.
- Hyndman, Rob J, and George Athanasopoulos. 2014. "Forecasting : Principles and Practice. Otexts; 2014." *Online at Http ://Otexts. Org/Fpp*.
- Jarque, Carlos M, and Anil K Bera. 1987. "A Test for Normality of Observations and Regression Residuals." *International Statistical Review/Revue Internationale de Statistique*, 163–72.
- Johansen, Søren. 1988. "Statistical Analysis of Cointegration Vectors." *Journal of Economic Dynamics and Control* 12 (2-3) : 231–54.
- Johansen, Soren, and Katarina Juselius. 1990. "Maximum Likelihood Estimation and Inference on Cointegration—with Appucations to the Demand for Money." *Oxford Bulletin of Economics and Statistics* 52 (2) : 169–210.
- Kilian, Lutz. 1998. "Small-Sample Confidence Intervals for Impulse Response Functions." *Review of Economics and Statistics* 80 (2) : 218–30.
- . 1999. "Finite-Sample Properties of Percentile and Percentile-t Bootstrap Confidence Intervals for Impulse Responses." *Review of Economics and Statistics* 81 (4) : 652–60.
- Kilian, Lutz, and Ufuk Demiroglu. 2000. "Residual-Based Tests for Normality in Autoregressions : Asymptotic Theory and Simulation Evidence." *Journal of Business & Economic Statistics* 18 (1) : 40–50.
- Lardic, Sandrine, and Valérie Mignon. 2002. *Econométrie Des séries Temporelles Macroéconomiques Et Financières*. Economica Paris.

- Ljung, Greta M, and George EP Box. 1978. "On a Measure of Lack of Fit in Time Series Models." *Biometrika* 65 (2) : 297–303.
- Lobato, Ignacio, John C Nankervis, and N Eugene Savin. 2001. "Testing for Autocorrelation Using a Modified Box-Pierce q Test." *International Economic Review* 42 (1) : 187–205.
- Lütkepohl, Helmut. 2005. *New Introduction to Multiple Time Series Analysis*. Springer Science & Business Media.
- Osterwald-Lenum, Michael. 1992. "A Note with Quantiles of the Asymptotic Distribution of the Maximum Likelihood Cointegration Rank Test Statistics 1." *Oxford Bulletin of Economics and Statistics* 54 (3) : 461–72.
- Pesaran, M Hashem, Yongcheol Shin, and Richard J Smith. 2001. "Bounds Testing Approaches to the Analysis of Level Relationships." *Journal of Applied Econometrics* 16 (3) : 289–326.
- Quinn, Barry G. 1980. "Order Determination for a Multivariate Autoregression." *Journal of the Royal Statistical Society : Series B (Methodological)* 42 (2) : 182–85.
- Schwarz, Gideon. 1978. "Estimating the Dimension of a Model." *The Annals of Statistics*, 461–64.
- Seabold, Skipper, and Josef Perktold. 2010. "Statsmodels : Econometric and Statistical Modeling with Python." In *9th Python in Science Conference*.
- Sheppard, Kevin. 2014. "Introduction to Python for Econometrics, Statistics and Data Analysis." University of Oxford.
- Sims, Christopher A. 1980. "Macroeconomics and Reality." *Econometrica : Journal of the Econometric Society*, 1–48.
- Toda, Hiro Y, and Taku Yamamoto. 1995. "Statistical Inference in Vector Autoregressions with Possibly Integrated Processes." *Journal of Econometrics* 66 (1-2) : 225–50.
- Tsay, Ruey S. 2013. *Multivariate Time Series Analysis : With r and Financial Applications*. John Wiley & Sons.
- Wright, Graham. 2018. "The Political Implications of American Concerns about Economic Inequality." *Political Behavior* 40 (2) : 321–43.
- Zellner, Arnold. 1962. "An Efficient Method of Estimating Seemingly Unrelated Regressions and Tests for Aggregation Bias." *Journal of the American Statistical Association* 57 (298) : 348–68.

Analyse des séries temporelles multivariées :

Théorie et application sous Python

Cet ouvrage présente les bases avancées de l'économétrie des séries temporelles. Il met l'accent sur les méthodes et les compétences indispensables à tout étudiant ou tout praticien désireux d'appliquer les méthodes avancées d'économétrie des séries temporelles. L'auteur met rapidement en pratique les divers concepts présentés avec des exemples appliqués sous le langage de programmation Python.

Ce manuel s'adresse aux étudiants de Licence et Master de mathématiques appliquées, d'économie et d'économétrie, ainsi qu'aux élèves des écoles d'ingénieurs. Il s'adresse également aux professionnels, praticiens de l'économétrie des séries temporelles.

Duvérier DJIFACK ZEBAZE est ingénieur en Data Science et gestion des risques de l'école nationale de la statistique et de l'analyse de l'information (ENSAI) de Rennes et Ingénieur Statisticien Economiste (ISE) de l'école nationale de la statistique et de l'analyse économique (ENSAE) de Dakar. Sa carrière débute en tant qu'ingénieur quantitatif au sein de l'équipe de modélisation de la Business Unit GLBA (Global Banking and Advisory) du groupe Société Générale.