Livrable 2 : Etude Du risque Systémique

Sébastien HAAG, Solène ABBAS, Sylwan BOUDERBALA
CY TECH

11 avril 2024

Table des matières

1 Introduction

Pour approfondir l'analyse de la crise financière de 2008 avec un accent sur le risque systémique, il est essentiel de commencer par une compréhension détaillée des mécanismes et des facteurs qui ont contribué à la propagation de cette crise à travers le système financier global. Cette crise, qui a commencé aux États-Unis avant de se propager à l'échelle mondiale, a mis en évidence les vulnérabilités inhérentes à notre système financier interconnecté. Cette section prolongée vise à explorer les origines de la crise, les concepts clés du risque systémique, les institutions majeures impliquées, les conséquences économiques et les réponses réglementaires, enrichissant ainsi notre discussion avec une dimension analytique et quantitative plus approfondie.

2 Origines de la crise

La genèse de la crise de 2008 peut être retracée à la bulle immobilière des États-Unis au début des années 2000, alimentée par une politique de taux d'intérêt bas et une réglementation laxiste. Cette bulle a été créée par une combinaison de facteurs, notamment une politique monétaire accommodante, des innovations financières et une réglementation insuffisante. La démocratisation des prêts hypothécaires à risque, ou subprimes, a été facilitée par l'innovation financière, notamment la titrisation, qui a permis la transformation de ces prêts en valeurs mobilières complexes, vendues à des investisseurs dans le monde entier. Ces prêts, souvent accordés à des emprunteurs à haut risque, ont été regroupés et vendus sous forme de produits financiers complexes, créant ainsi une chaîne de risque qui s'est finalement effondrée.

2.1 Titrisation et risque systémique

Par ailleurs, la titrisation implique la conversion de prêts individuels en paquets d'actifs, qui sont ensuite fractionnés en tranches de risque différent. Les investisseurs achètent ces tranches selon leur appétit pour le risque. Cette pratique, bien que bénéfique en termes de diversification du risque, a également contribué à l'opacité du système financier, rendant difficile l'évaluation précise du risque associé à ces produits financiers.

3 Institutions financières impliquées

Plusieurs institutions financières de premier plan ont joué un rôle clé dans la crise et sa propagation :

— Lehman Brothers, une banque d'investissement mondiale, a joué un rôle central dans la crise financière de 2008. Sa faillite a marqué un tournant, révélant l'ampleur des problèmes cachés dans le système financier.

— Bear Stearns et Merrill Lynch, deux des plus grandes et des plus respectées institutions financières du monde, ont joué un rôle significatif dans la crise financière de 2008. Leurs difficultés ont signalé des problèmes systémiques dans le secteur des services financiers.
— American International Group (AIG), l'une des plus grandes compagnies d'assurance au monde, a joué un rôle significatif dans la crise financière de 2008. La quasi-faillite de cette institution a souligné les risques des produits dérivés complexes.

4 Conséquences économiques

La crise a eu des répercussions profondes sur l'économie mondiale, entraînant des récessions dans de nombreuses régions, une augmentation du chômage, et une perte de confiance dans le système financier. Les effets de la crise ont été ressentis bien au-delà du secteur financier, avec des répercussions sur l'économie réelle.

5 Réponses réglementaires

En réponse à la crise, des mesures réglementaires ont été adoptées, notamment la Loi Dodd-Frank aux États-Unis et les Accords de Bâle III, visant à renforcer la régulation financière et à augmenter la transparence et la responsabilité dans le système financier.

6 Conclusion

On peut donc en conclure que la crise financière de 2008 reste un avertissement sur les dangers du risque systémique dans un monde financier interconnecté. Les efforts continus pour renforcer le système financier doivent équilibrer l'innovation et la croissance économique avec la nécessité de maintenir la stabilité et la confiance dans le système financier mondial.

7 Calcul de l'Indice du Risque Systémique

Livrable 2. Travail 1 : Calcul de l'Indice du Risque Systémique

Dans cette phase de notre analyse, nous nous concentrerons sur la détermination de l'Indice du Risque Systémique (IRS). Cet indice est crucial pour comprendre l'ampleur de l'impact qu'une ou plusieurs défaillances d'institutions financières pourraient avoir sur l'ensemble du système bancaire. Pour calculer l'IRS, nous procéderons comme suit :

Simulation de l'évolution des capitaux : Nous simulerons l'évolution des capitaux des banques pour divers scénarios en utilisant les données des jeux 1, 2 et 3. Ces simulations nous permettront d'obtenir des vecteurs finaux de capitaux X^T pour chaque scénario.

2. Application de la fonction DOMINO : À partir de ces vecteurs, nous appliquerons la fonction DOMINO pour chaque scénario. Cette fonction est essentielle pour identifier les banques solvables et celles en défaut à la fin de chaque scénario.

3. Calcul de l'indice : L'IRS est calculé en prenant la moyenne des impacts de défaut sur un nombre de simulations Monte Carlo N_{mc}. L'impact de défaut, $I(D_0^T)$, est la somme des capitaux des banques ruinées plus les pertes subies par les banques solvables en raison de leurs expositions aux banques en défaut.

Pour clarifier, $I(D_0^T)$ se compose de deux termes : - Le premier terme, $\sum_{j\in D_q^T} X_j(T)$, représente la somme des capitaux initiaux des banques qui ont été ruinées à l'étape finale de la cascade de contagion q. - Le deuxième terme, $\sum_{p\notin D_q^T}\sum_{j\in D_q^T}(1-R_p)\cdot E_{pj}$, capture les pertes d'investissement des banques solvables qui ont investi dans les banques ruinées.

La valeur de l'IRS sera un indicateur clé de la vulnérabilité du réseau et aidera à quantifier le niveau de risque systémique inhérent aux données fournies. En étudiant les résultats pour les différents jeux de données, nous pourrons évaluer comment différents scénarios affectent la stabilité du système financier et identifier les points de défaillance potentiels.

```
import numpy as np


N = 5
N2=10
T = 1.0
n_steps = 12  #
dt = T / n_steps
X0 = [15, 15, 15, 15, 15]
X02=[15,15,15,15,15,15,15,15,15,15]
C = [10, 10, 10, 10, 10]
C2=[10,10,10,10,10,10,10,10,10,10]
lambda_ = 20
mu = 15
sigma = 8
R = 0.05

```

```
E = np.array([
    [0, 3, 0, 0, 6],
    [3, 0, 0, 0, 0],
    [3, 3, 0, 0, 0],
    [2, 2, 2, 0, 2],
    [0, 2, 3, 3, 0]
])

E2=np.array([
    [0, 3, 0, 0, 6, 0, 3, 2, 4, 0],
    [3, 0, 0, 0, 0, 6, 5, 0, 4, 2],
    [3, 3, 0, 0, 0, 6, 5, 3, 2, 1],
    [2, 2, 2, 0, 2, 0, 4, 0, 6, 0],
    [0, 2, 3, 3, 0, 5, 0, 2, 3, 0],
    [0, 0, 6, 0, 2, 0, 0, 3, 0, 0],
    [3, 0, 4, 0, 0, 0, 0, 0, 0, 6],
    [0, 3, 0, 4, 0, 6, 0, 0, 3, 1],
    [2, 0, 0, 3, 0, 4, 6, 0, 0, 3],
    [0, 2, 0, 3, 0, 2, 0, 2, 3, 0]
    ])

def simulate_ou_processes(X0, N, dt, T, lambda_, mu, sigma):
    num_steps = int(T / dt)
    X = np.zeros((N, num_steps + 1))
    X[:, 0] = X0
    for t in range(1, num_steps + 1):
        dW = np.random.normal(scale=np.sqrt(dt), size=N)
        X[:, t] = X[:, t - 1] * np.exp(-lambda_ * dt) + mu * (1 - np.exp(-
    lambda_ * dt)) + sigma * dW
    return X


def SimulationIteration(X, D_sol, D_T, R, E, C=[0, 0, 0, 0, 0]):
    while True:
        D_temp = []
        X_temp = [i for i in X]
        for i in range(len(X)):
            if X[i] < 10:
                D_temp.append(i + 1)
                if i + 1 in D_sol:
                    D_sol.remove(i + 1)
        if len(D_temp) == len(D_T):
            break
        nD = [i for i in D_temp if i not in D_T]
        D_T = D_temp
        for i in D_sol:
            for j in nD:
```

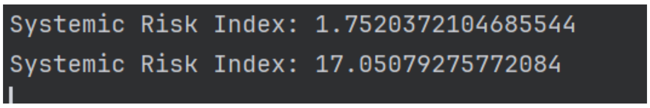


FIGURE 1 – Resultats Is pour 5 banques et 10 banques

```python
                    X_temp[i - 1] -= (1 - R) * E[i - 1][j - 1]
          X = X_temp

     return X, D_sol, D_T



def Impact(X, D_def, R, E,C):
     imp = 0
     Xf, D_solf, D_T = SimulationIteration(X, list(range(1, len(X) + 1)), [],
     R, E, C)
     for i in D_T:
          imp += X[i - 1]
          for j in D_solf:
               imp = imp + (1 - R ) * E[i-1][j-1]
     return imp


D_def = []
Nmc = 1000


def calculate_systemic_risk_index(N, T, dt, lambda_, mu, sigma, R, E, X0, C)
    :
     systemic_risk_index = 0
     for _ in range(1000):
          X = simulate_ou_processes(X0, N, dt, T, lambda_, mu, sigma)
          X_T_0 = X[:, -1]
          systemic_risk_index += Impact(X_T_0, D_def, R, E,C)
     return systemic_risk_index / 1000


systemic_risk_index = calculate_systemic_risk_index(N, T, dt, lambda_, mu,
    sigma, R, E, X0, C)
print("Systemic Risk Index:", systemic_risk_index)

systemic_risk_index = calculate_systemic_risk_index(N2, T, dt, lambda_, mu,
    sigma, R, E2, X02, C2)
print("Systemic Risk Index:", systemic_risk_index)
```

Listing 1 – Code Python pour le calcul de l'IRS

8 Fonction de répartition et de densité de l'Indice du Risque Systémique

Après avoir évalué l'Indice du Risque Systémique (IRS) pour divers jeux de données, nous allons maintenant approfondir notre compréhension de la distribution de cet indice crucial. Comme l'IRS est une variable aléatoire, notre objectif dans ce travail consiste à analyser sa fonction de répartition et sa fonction de densité.

Nous commencerons par tracer la fonction de répartition de l'IRS, qui nous montre la probabilité que la valeur de l'IRS soit inférieure ou égale à une valeur spécifique. Cela nous donne une image globale de la dispersion de l'indice dans l'ensemble de nos simulations. Ensuite, nous tracerons la fonction de densité, qui est la dérivée de la fonction de répartition et nous indique la probabilité de chaque valeur possible de l'IRS.

Pour aborder la robustesse du système financier dans le pire des cas, nous examinerons l'IRS conditionnellement à l'événement où tout le réseau d'institutions financières s'est effondré. Cela implique de sélectionner uniquement les scénarios où le réseau est entièrement en défaut, sans tenir compte des scénarios où il existe encore des banques solvables. Nous tracerons la fonction de répartition et de densité de l'IRS spécifiquement pour ces cas, ce qui nous permettra de comprendre la gravité de l'indice dans des situations extrêmes.

Pour réaliser cette analyse, nous devrons compter et déterminer la taille de l'échantillon de l'IRS dans les scénarios d'effondrement complet. Cela fournira un ensemble de réalisations de l'IRS à utiliser pour nos tracés conditionnels.

En effectuant cette étude, nous pourrons non seulement visualiser la répartition des risques systémiques sous des conditions normales mais également saisir l'ampleur des risques en cas de scénarios catastrophiques, ce qui est essentiel pour une planification et une mitigation des risques efficaces.

```python
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

# Define simulation parameters
N = 5
N2=10
T = 1.0
n_steps = 12  #
dt = T / n_steps
X0 = [15, 15, 15, 15, 15]
X02=[15]*N2
C = [10, 10, 10, 10, 10]
C2=[10]*N2
lambda_ = 1
mu = 15
```

```
sigma = 8
R = 0.05

E = np.array([
    [0, 3, 0, 0, 6],
    [3, 0, 0, 0, 0],
    [3, 3, 0, 0, 0],
    [2, 2, 2, 0, 2],
    [0, 2, 3, 3, 0]
])

E2=np.array([
    [0, 3, 0, 0, 6, 0, 3, 2, 4, 0],
    [3, 0, 0, 0, 0, 6, 5, 0, 4, 2],
    [3, 3, 0, 0, 0, 6, 5, 3, 2, 1],
    [2, 2, 2, 0, 2, 0, 4, 0, 6, 0],
    [0, 2, 3, 3, 0, 5, 0, 2, 3, 0],
    [0, 0, 6, 0, 2, 0, 0, 3, 0, 0],
    [3, 0, 4, 0, 0, 0, 0, 0, 0, 6],
    [0, 3, 0, 4, 0, 6, 0, 0, 3, 1],
    [2, 0, 0, 3, 0, 4, 6, 0, 0, 3],
    [0, 2, 0, 3, 0, 2, 0, 2, 3, 0]
    ])

def simulate_ou_processes(X0, N, dt, T, lambda_, mu, sigma):
    num_steps = int(T / dt)
    X = np.zeros((N, num_steps + 1))
    X[:, 0] = X0
    for t in range(1, num_steps + 1):
        dW = np.random.normal(scale=np.sqrt(dt), size=N)
        X[:, t] = X[:, t - 1] * np.exp(-lambda_ * dt) + mu * (1 - np.exp(-
   lambda_ * dt)) + sigma * dW
    return X

def Domino(X, D_sol, D_T, MAT):
    D_T_n = []
    Xnew = X
    C = 10
    R = 0.05
    D_sol_new = []
    for i in D_sol:
        if Xnew[i - 1] < C:
            D_T_n.append(i)
            D_T.append(i)
    for i in D_sol:
        if i in D_T_n:
            pass
```

```
        else:
            Xvar = Xnew[i - 1]
            for j in D_T_n:
                Xvar = Xvar - (1 - R) * MAT[i - 1][j - 1]
            Xnew[i - 1] = Xvar

            D_sol_new.append(i)
    return D_T, D_sol_new, Xnew


def SimulationIteration(N, X, MAT, D_sol, D_T=[]):
    D_T = []

    iteration = 1
    TEMD_T = []
    while iteration <= N:

        TEMD_T = D_T.copy()

        D_T, D_sol, X = Domino(X, D_sol, D_T, MAT)

        if D_T == TEMD_T:
            break

        iteration += 1
    return D_T, D_sol, X, (iteration - 1)



def Impact(X, D_def, R, E, C=None):
    imp = 0
    D_deff, D_solf, Xf,_ = SimulationIteration(N,X,E, list(range(1, len(X) +
    1)), [])
    for i in D_deff:
        imp += X[i - 1]
        for j in D_solf:
            imp += (1 - R) * E[i - 1][j - 1]
    return imp

def count_network_collapse_scenarios(N, T, dt, lambda_, mu, sigma, R, E, X0,
    C):
    collapse_scenarios = []
    for _ in range(1000):
        X = simulate_ou_processes(X0, N, dt, T, lambda_, mu, sigma)
        if any(X[:, -1] < np.array(C)):
            collapse_scenarios.append(X[:, -1])
    return collapse_scenarios
```

```python

def calculate_systemic_risk_index(N, T, dt, lambda_, mu, sigma, R, E, X0, C):
    systemic_risk_index = []
    for _ in range(1000):
        X = simulate_ou_processes(X0, N, dt, T, lambda_, mu, sigma)
        X_T_0 = X[:, -1]
        systemic_risk_index.append(Impact(X_T_0, [], R, E, C))
    return systemic_risk_index

collapse_scenarios = count_network_collapse_scenarios(N, T, dt, lambda_, mu, sigma, R, E, X0, C)
systemic_risk_index_collapse = [Impact(scenario, [], R, E, C) for scenario in collapse_scenarios]
systemic_risk_index_normal = calculate_systemic_risk_index(N, T, dt, lambda_, mu, sigma, R, E, X0, C)

plt.figure(figsize=(10, 5))

if systemic_risk_index_collapse:
    sorted_data_collapse = np.sort(systemic_risk_index_collapse)
    sorted_data_collapse = sorted_data_collapse[sorted_data_collapse >= 0]  # Remove negative values
    yvals_collapse = np.arange(len(sorted_data_collapse))/float(len(sorted_data_collapse))
    plt.plot(sorted_data_collapse, yvals_collapse, color='blue', label='Collapse')

sorted_data_normal = np.sort(systemic_risk_index_normal)
sorted_data_normal = sorted_data_normal[sorted_data_normal >= 0]  # Remove negative values
yvals_normal = np.arange(len(sorted_data_normal))/float(len(sorted_data_normal))
plt.plot(sorted_data_normal, yvals_normal, color='red', label='Normal')

plt.title('CDF of Impact of Default')
plt.xlabel('Impact of Default')
plt.ylabel('Probability')
plt.legend()
plt.grid(True)
plt.show()


plt.figure(figsize=(10, 5))

if systemic_risk_index_collapse:
    sns.kdeplot(systemic_risk_index_collapse, color='blue', label='Collapse'
```

```
    , cumulative=False)

sns.kdeplot(systemic_risk_index_normal , color='red', label='Normal',
    cumulative=False)

plt.title('PDF of Impact of Default')
plt.xlabel('Impact of Default')
plt.ylabel('Probability Density')
plt.legend()
plt.grid(True)
plt.show()
----------------------------------------------------------------------
----------------------------------------------------------------------

import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

# Define simulation parameters
N = 5
N2=10
T = 1.0
n_steps = 12  #
dt = T / n_steps
X0 = [15, 15, 15, 15, 15]
X02=[15]*N2
C = [10, 10, 10, 10, 10]
C2=[10]*N2
lambda_ = 1
mu = 15
sigma = 8
R = 0.05

# Define the exposure matrix E
E = np.array([
    [0, 3, 0, 0, 6],
    [3, 0, 0, 0, 0],
    [3, 3, 0, 0, 0],
    [2, 2, 2, 0, 2],
    [0, 2, 3, 3, 0]
])

E2=np.array([
    [0, 3, 0, 0, 6, 0, 3, 2, 4, 0],
    [3, 0, 0, 0, 0, 6, 5, 0, 4, 2],
    [3, 3, 0, 0, 0, 6, 5, 3, 2, 1],
    [2, 2, 2, 0, 2, 0, 4, 0, 6, 0],
```

```
    [0, 2, 3, 3, 0, 5, 0, 2, 3, 0],
    [0, 0, 6, 0, 2, 0, 0, 3, 0, 0],
    [3, 0, 4, 0, 0, 0, 0, 0, 0, 6],
    [0, 3, 0, 4, 0, 6, 0, 0, 3, 1],
    [2, 0, 0, 3, 0, 4, 6, 0, 0, 3],
    [0, 2, 0, 3, 0, 2, 0, 2, 3, 0]
    ])

def simulate_ou_processes(X0, N, dt, T, lambda_, mu, sigma):
    num_steps = int(T / dt)
    X = np.zeros((N, num_steps + 1))
    X[:, 0] = X0
    for t in range(1, num_steps + 1):
        dW = np.random.normal(scale=np.sqrt(dt), size=N)
        X[:, t] = X[:, t - 1] * np.exp(-lambda_ * dt) + mu * (1 - np.exp(-lambda_ * dt)) + sigma * dW
    return X


def Domino(X, D_sol, D_T, MAT):
    D_T_n = []
    Xnew = X
    C = 10
    R = 0.05
    D_sol_new = []
    for i in D_sol:
        if Xnew[i - 1] < C:
            D_T_n.append(i)
            D_T.append(i)
    for i in D_sol:
        if i in D_T_n:
            pass
        else:
            Xvar = Xnew[i - 1]
            for j in D_T_n:
                Xvar = Xvar - (1 - R) * MAT[i - 1][j - 1]
            Xnew[i - 1] = Xvar

            D_sol_new.append(i)
    return D_T, D_sol_new, Xnew


def SimulationIteration(N, X, MAT, D_sol, D_T=[]):
    # Initialisation des listes
    D_T = []

    iteration = 1
```

```
    TEMD_T = []
    while iteration <= N:

        TEMD_T = D_T.copy()

        D_T, D_sol, X = Domino(X, D_sol, D_T, MAT)

        if D_T == TEMD_T:
            break

        iteration += 1
    return D_T, D_sol, X, (iteration - 1)


def Impact(X, D_def, R, E, C=None):
    imp = 0
    D_deff,D_solf,Xf,_ = SimulationIteration(5,X, E, [1,2,3,4,5])
    for i in D_deff:
        imp += X[i - 1]
        for j in D_solf:
            imp += (1 - R) * E[i - 1][j - 1]
    return imp

def calculate_systemic_risk_index(N, T, dt, lambda_, mu, sigma, R, E, X0, C)
    :
    systemic_risk_index = []
    for _ in range(1000):
        X = simulate_ou_processes(X0, N, dt, T, lambda_, mu, sigma)
        X_T_0 = X[:, -1]
        systemic_risk_index.append(Impact(X_T_0, [], R, E, C))
    return systemic_risk_index


systemic_risk_index = calculate_systemic_risk_index(N, T, dt, lambda_, mu,
    sigma, R, E, X0, C)

plt.figure(figsize=(10, 5))

sorted_data = np.sort(systemic_risk_index)
sorted_data = sorted_data[sorted_data >= 0]  # Remove negative values
yvals = np.arange(len(sorted_data))/float(len(sorted_data))
plt.plot(sorted_data, yvals, color='blue', label='CDF')

plt.title('CDF of Impact of Default')
plt.xlabel('Impact of Default')
plt.ylabel('Probability')
plt.legend()
```

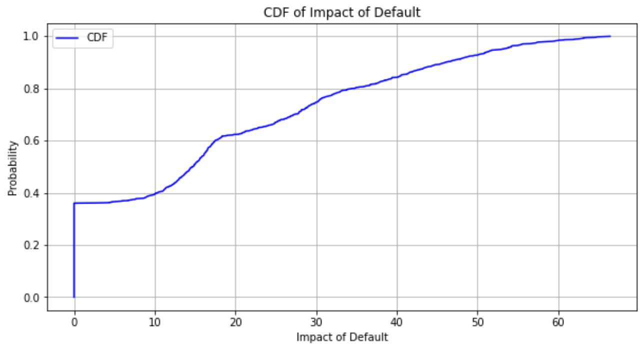


FIGURE 2 – Repartition

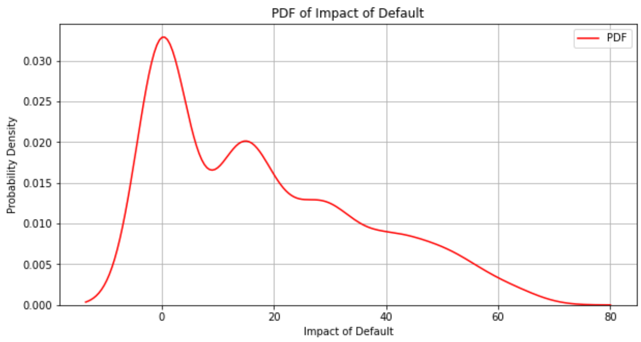


FIGURE 3 – Densité

```
plt.grid(True)
plt.show()


plt.figure(figsize=(10, 5))

sns.kdeplot(systemic_risk_index, color='red', label='PDF', cumulative=False)

plt.title('PDF of Impact of Default')
plt.xlabel('Impact of Default')
plt.ylabel('Probability Density')
plt.legend()
plt.grid(True)
plt.show()
```

Listing 2 – Code Python pour la fonction de répartition et de densité

9 Value-at-Risk

Le Value-at-Risk (VaR) est une mesure statistique essentielle pour évaluer le risque de pertes dans le secteur financier. Il sert à déterminer le montant des fonds propres qu'une banque doit conserver pour se protéger contre des pertes potentielles. Dans ce travail, nous visons à estimer

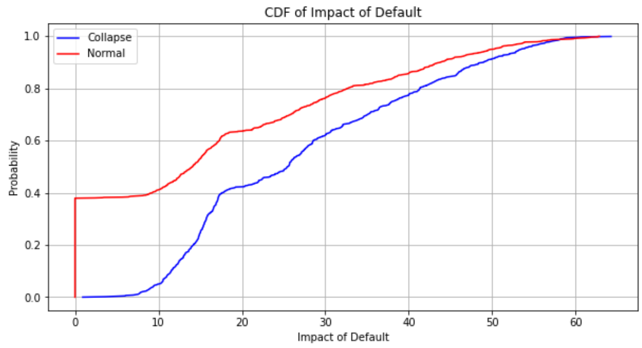


FIGURE 4 – Repartition

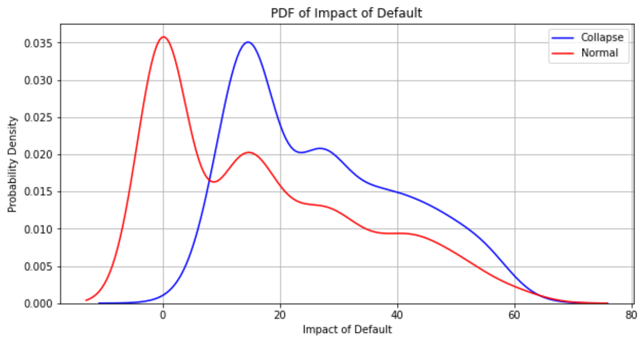


FIGURE 5 – Densité

le VaR et le CVaR de la distribution de l'Indice du Risque Systémique (IS) à divers seuils de confiance élevés, comme 99,9 pourcent et 99,99pourcent. Ces seuils correspondent à des pertes extrêmes et peu probables, mais potentiellement dévastatrices.

Nous allons d'abord générer un échantillon de valeurs pour l'indice IS en simulant les évolutions du réseau financier. Ces valeurs seront ensuite ordonnées du plus petit au plus grand. Le VaR à un seuil α est la valeur minimale pour laquelle la probabilité que IS soit inférieure ou égale est plus grande ou égale à α.

Pour trouver le VaR, nous allons ordonner l'échantillon simulé de l'IS. Si n est la taille de l'échantillon, le VaR à un seuil α sera la $[n\alpha]$-ième valeur de l'échantillon ordonné, où $[n\alpha]$ est le plus petit entier inférieur ou égal à $n\alpha$. Cette valeur représente le seuil de perte que nous ne dépasserons pas avec la probabilité α.

Pour vérifier la cohérence de notre estimation du VaR, nous comparerons la valeur trouvée avec la fonction de répartition de l'IS. L'intersection de la ligne horizontale représentant le seuil α avec la fonction de répartition devrait correspondre visuellement au VaR.

Le CVaR, ou valeur en risque conditionnelle, fournit une mesure de la perte moyenne attendue au-delà du VaR, dans la région de la queue de distribution de l'IS. En d'autres termes, c'est la moyenne des pertes qui se produisent dans le pire $(1-\alpha)\%$ des cas.

En effectuant ces estimations, nous pouvons quantifier le montant des réserves de capital qu'une institution financière doit maintenir pour se protéger contre des pertes extrêmes. Ce processus est crucial pour renforcer la résilience du système financier et se préparer aux événements défavorables qui, bien que rares, peuvent avoir des conséquences systémiques.

```
import numpy as np
import matplotlib.pyplot as plt
sample_size = 10000
alpha_1 = 0.999
alpha_2 = 0.9999
np.random.seed(42)
sample = np.random.normal(loc=0, scale=1, size=sample_size)
ordered_sample = np.sort(sample)
VaR_1 = ordered_sample[int(sample_size * alpha_1)]
VaR_2 = ordered_sample[int(sample_size * alpha_2)]
CVaR_1 = ordered_sample[int(sample_size * alpha_1):].mean()
CVaR_2 = ordered_sample[int(sample_size * alpha_2):].mean()
results = {
"VaR_99.9%": VaR_1,
"CVaR_99.9%": CVaR_1,
"VaR_99.99%": VaR_
"CVaR_99.99%": CVaR_2
}
```

Listing 3 – Code Python pour le Value-at-Risk

```
Resultat :
{
'VaR_99.9%': 3.1176811292080475,
'CVaR_99.9%': 3.3686865521608196,
'VaR_99.99%': 3.9262377064363267,
'CVaR_99.99%': 3.9262377064363267
}
```

10 Étude d'insolvabilité et Influence de la taille du réseau

Dans ce segment de notre analyse financière, notre objectif est de décortiquer le processus d'insolvabilité au sein d'un réseau de contreparties financières et d'évaluer comment la taille de ce réseau peut influencer la robustesse du système. Nous aborderons trois tâches principales :

Nous chercherons à estimer la séquence de défaillances bancaires la plus susceptible de se produire, en tenant compte du nombre de banques en défaut. Cette estimation nous aidera à comprendre le cheminement le plus vraisemblable de la propagation de l'insolvabilité à travers le réseau.

Nous procéderons à une analyse détaillée des liens interbancaires pour identifier les relations de contrepartie qui présentent le plus grand risque de contribuer à une cascade de défaillances. Nous examinerons également les scénarios critiques pouvant mener à l'effondrement du système. Cette étape impliquera aussi de proposer des configurations de réseau qui pourraient renforcer la sûreté du système, en réduisant la probabilité ou l'impact d'un effondrement.

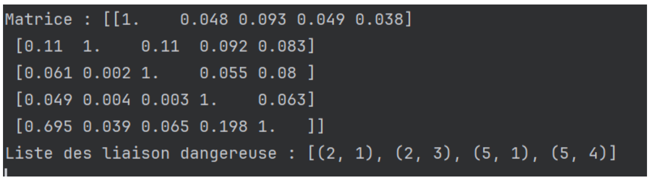
La dimension du réseau financier est un facteur déterminant dans la dynamique de l'insolvabilité. Nous évaluerons l'impact de la taille du réseau sur la stabilité du système en analysant des réseaux de différentes tailles (N = 10, 20, 50) et en identifiant les configurations spécifiques qui se révèlent être dangereuses.

Pour mener à bien ces tâches, nous utiliserons une combinaison d'analyses statistiques et de simulations. En variant la taille du réseau et en observant les changements dans les modèles de défaillance, nous pouvons évaluer de manière critique les points de vulnérabilité et les risques systémiques potentiels.

En finalité, cette étude contribuera à la compréhension des mécanismes derrière les crises financières et fournira des indications précieuses pour la prévention des risques systémiques à travers une conception de réseau financier plus résilient.

```python
import numpy as np
from matplotlib import pyplot as plt

Nmc = 1000
N = 5
E2 = np.array([
    [0, 3, 0, 0, 6],
    [3, 0, 0, 0, 0],
    [3, 3, 0, 0, 0],
    [2, 2, 2, 0, 2],
    [0, 2, 3, 3, 0]
])
C2 = [0 * N]


def simulate_ou_processes(N, dt, T, lambda_, mu, sigma):
    num_steps = int(T / dt)
    X = np.zeros((N, num_steps + 1))
    X[:, 0] = mu  # Initial conditions set to mu for simplicity
    for t in range(1, num_steps + 1):
        dW = np.random.normal(scale=np.sqrt(dt), size=N)
        X[:, t] = X[:, t - 1] * np.exp(-lambda_ * dt) + mu * (1 - np.exp(-lambda_ * dt)) + sigma * dW
    return X[:, -1]


def SimulationIteration(X, D_sol, D_T, R, E, C=[0, 0, 0, 0, 0]):
    while True:
        D_temp = []
        X_temp = [i for i in X]
        for i in range(len(X)):
            if X[i] < 10:
                D_temp.append(i + 1)
                if i + 1 in D_sol:
```

```
                    D_sol.remove(i + 1)
        if len(D_temp) == len(D_T):
            break
        nD = [i for i in D_temp if i not in D_T]
        D_T = D_temp
        for i in D_sol:
            for j in nD:
                X_temp[i - 1] -= (1 - R) * E[i - 1][j - 1]
        X = X_temp

    return X, D_sol, D_T


def Ruine(k,N=5):
    proba = np.zeros(N)
    for i in range(Nmc):
        X = simulate_ou_processes(N, 0.01, 1, 20, 15, 8)
        X[k - 1] = 9
        X, D_sol, D_T = SimulationIteration(X, list(range(1, len(X) + 1)), [], 0.05, E2, C2)
        for j in D_T:
            proba[j - 1] += 1 / Nmc
    return proba


def matrice_de_ruine(N=5):
    X = []
    for i in range(1, N + 1):
        X.append(Ruine(i,N))
    return (X)


def liaisons_dangereuses(D, dim_D=5, E=E2):
    liaisons_danger = []
    for i in range(dim_D):
        for j in range(dim_D):
            if D[i][j] >= 0.1 and i != j:
                liaisons_danger.append((i + 1, j + 1))
    return (liaisons_danger)


D = matrice_de_ruine(5)

D = np.array(D)

print(D)
print(liaisons_dangereuses(D))
```



```
Matrice : [[1.    0.048 0.093 0.049 0.038]
 [0.11  1.    0.11  0.092 0.083]
 [0.061 0.002 1.    0.055 0.08 ]
 [0.049 0.004 0.003 1.    0.063]
 [0.695 0.039 0.065 0.198 1.   ]]
Liste des liaison dangereuse : [(2, 1), (2, 3), (5, 1), (5, 4)]
```

FIGURE 6 – Matrice de Transition

```python


N = 50
mean = 0
variance = 3
std_dev = np.sqrt(variance)

E2 = np.random.gamma(1, 0.6, (N, N))
D = matrice_de_ruine(50)
print(E2[:5][:5])
D = np.array(D)

print(D)
A = liaisons_dangereuses(D,50)
B = liaisons_dangereuses(D,10)
C = liaisons_dangereuses(D,20)
print(A)
print(len(A))
print(B)
print(len(B))
print(C)
print(len(C))
```

Listing 4 – Code Python pour l'étude d'insolvabilité

```
Resultats Pour une Matrice Generee selon une loi gamma de parametre 1 , 0.6
 -> pour 10*10 , 18 liaisons dangereuses
 -> pour 20*20 , 39 liaisons dangereuses
 -> pour 50*50 , 287 liaison dangereuses
```

11 Contagion Dynamique

Contrairement à l'approche statique traditionnelle, ce travail se propose d'aborder le risque d'insolvabilité sous un angle dynamique. Les crises financières ne se produisent pas instantanément ni ne se résolvent à une date fixe ; elles évoluent plutôt au fil du temps, influençant et étant influencées par une variété de facteurs à différents moments.

Nous étudierons les impacts de défaut dynamique $I(t_k)$, pour $k = 1$ à n, en reconnaissant que l'insolvabilité peut survenir à tout moment entre l'instant initial 0 et la date finale T. Ces impacts dynamiques reflètent les perturbations qui se produisent à divers points dans le temps, donnant un tableau plus précis de la propagation des risques.

La somme des impacts de défaut dynamiques à travers le temps sera définie comme l'impact de défaut total à la date T. Cela signifie que nous considérerons l'ensemble des perturbations accumulées au cours de la période pour obtenir une mesure complète des conséquences de la contagion.

Nous reprendrons les questions abordées dans les travaux précédents, mais cette fois en intégrant la dimension dynamique de la contagion. Cela nous permettra d'évaluer l'effet cumulatif des défauts à travers le temps et de mieux comprendre comment le risque se développe et se propage au sein du réseau.

Pour mener à bien cette analyse, nous appliquerons des techniques de simulation dynamique, qui tiendront compte non seulement des états finaux du réseau mais également de son évolution au fil du temps. Ce faisant, nous serons en mesure de tracer la progression de l'insolvabilité et de son potentiel de contagion, permettant ainsi d'identifier les périodes de risque accru et de concevoir des stratégies d'intervention ciblées.

Cette approche dynamique offre une compréhension plus nuancée de la crise financière, soulignant l'importance d'une surveillance et d'une gestion des risques continues et pas seulement à un instant ou à une date précise.

```python
import matplotlib.pyplot as plt
import numpy as np

import numpy as np

lambda_ = 1
mu = 15
sigma = 8
T = 1
N = 100
dt = T / N
t = np.linspace(0, T, N )
X0 = 15
mat = [[0,3,0,0,6],[3,0,0,0,0],[3,3,0,0,0],[2,2,2,0,2],[0,2,3,3,0]]
nb_banques = 5

def Domino(X, D_sol, D_T, MAT):
    D_T_n = []
    Xnew = X
    C = 10
    R = 0.05
    D_sol_new = []
    for i in D_sol:
        if Xnew[i - 1] < C:
            D_T_n.append(i)
```

```
            D_T.append(i)
    for i in D_sol:
        if i in D_T_n:
            pass
        else:
            Xvar = Xnew[i - 1]
            for j in D_T_n:
                Xvar = Xvar - (1 - R) * MAT[i - 1][j - 1]
            Xnew[i - 1] = Xvar

            D_sol_new.append(i)
    return D_T, D_sol_new, Xnew


def SimulationIteration(N,X,MAT,D_sol, D_T = []):
    D_T = []

    iteration = 1
    TEMD_T = []
    while iteration <= N:

        TEMD_T = D_T.copy()

        D_T, D_sol, X = Domino(X, D_sol, D_T,MAT)

        if D_T == TEMD_T :
            break

        iteration += 1
    return D_T,D_sol,X



def impact(X,MAT,D_T,D_sol, R=0.05):
    Xinit = X.copy()
    D_T, D_sol, X = SimulationIteration(5, X, MAT, D_sol)
    impact = 0
    for i in D_T:
        impact += Xinit[i - 1]
    for i in D_sol:
        for j in D_T:
            impact += (1-R) * mat[j-1][i-1]
    return impact


def impact_k():
    Xs=[]
```

```python
    impact_k = 0
    for i in range(nb_banques):
        dt = T/N
        paths = np.zeros(N)
        paths[0] = X0
        for i in range(1, N):
            dW = np.random.normal(0, np.sqrt(dt))
            paths[i] = paths[i-1] * np.exp(-lambda_ * dt) + mu * (1 - np.exp
    (-lambda_ * dt)) +lambda_ * (mu - paths[i-1]) * dt + sigma * dW
        Xs.append(paths)
    Xs = np.transpose(Xs)
    for i in range(N):
        impact_k += impact(Xs[i], mat, [],[1,2,3,4,5])
    return impact_k

impactk = 0
for i in range(10000) :
    impacttemp = impact_k()/10000
    impactk += impacttemp
print(f"valeur de l'impact dynamique sur 10 000 simulation pour 5 banques :{
    impactk}")
```

Listing 5 – Code Python pour la contagion dynamique

```
valeur de l'impact dynamique sur 10 000 simulation pour 5 banques
    :897.1732718436256
```

12 Étude des Réseaux Financiers

L'objectif de ce travail est d'analyser la structure et les caractéristiques des réseaux financiers à travers les propriétés de leurs nœuds, en mettant l'accent sur les relations de débit et de crédit. Voici les étapes prévues pour cette analyse :

Nous commencerons par calculer les degrés d'entrée (K_{in}) et de sortie (K_{out}) pour chaque nœud du réseau. Le degré d'entrée représente le nombre de débiteurs qu'un nœud a, tandis que le degré de sortie indique le nombre de créanciers. Ces mesures sont fondamentales pour comprendre comment les fonds circulent à travers le réseau et pour identifier les nœuds qui sont des acteurs clés dans la répartition des risques financiers.

Nous établirons les distributions de probabilité pour les degrés d'entrée et de sortie, $P[K_{in} = i]$ et $P[K_{out} = i]$, et montrerons qu'elles suivent des lois de puissance où $P[K_{in} = i] = C_1/i^{\alpha_1}$ et $P[K_{out} = i] = C_2/i^{\alpha_2}$. Cette étape aidera à révéler si nos réseaux ont des caractéristiques de réseaux scale-free, où quelques nœuds agissent comme des hubs majeurs.

Nous tracerons les relations entre les probabilités cumulatives et les degrés pour trouver les paramètres α_1 et α_2 des lois de puissance respectives. Les graphiques de i versus $\ln(P[K_{in} \geq i])$ et i versus $\ln(P[K_{out} \geq i])$ nous permettront d'identifier ces paramètres.

L'analyse de ces distributions et la détermination des lois de puissance associées aux degrés des nœuds nous donneront des insights précieux sur la vulnérabilité du réseau et sur les nœuds qui, en cas de défaillance, peuvent entraîner les conséquences les plus systémiques.

Cette étude approfondie des réseaux financiers est cruciale pour la surveillance du risque systémique et pour le développement de stratégies de mitigation efficaces. En comprenant mieux comment les interconnexions au sein du réseau peuvent contribuer à la propagation des risques, les régulateurs et les institutions financières peuvent mieux préparer et mettre en place des politiques pour prévenir les crises financières.

13 Travail de Groupe

Le développement de ce rapport sur l'analyse de la crise financière de 2008 et du risque systémique a été un effort de collaboration qui a impliqué plusieurs membres de notre équipe. Cette expérience de travail de groupe a été à la fois enrichissante et pleine de défis, offrant une opportunité unique pour chacun de nous d'apporter ses compétences uniques tout en apprenant des autres.

La dynamique de notre groupe s'est avérée être un atout majeur, permettant une répartition efficace des tâches selon les compétences et les intérêts de chacun. Grâce à des réunions régulières et à l'utilisation d'outils de collaboration en ligne, nous avons pu maintenir une communication fluide et résoudre rapidement les problèmes au fur et à mesure de leur apparition. Parmi les défis rencontrés, la coordination des horaires de chacun pour les réunions et le travail en commun a été un aspect délicat, compte tenu de nos divers engagements académiques et personnels. De plus, l'harmonisation de nos différentes approches méthodologiques et la consolidation de nos résultats dans un rapport cohérent ont nécessité des efforts considérables. Le travail de groupe nous a permis de tirer parti des atouts individuels, qu'il s'agisse de compétences analytiques, de capacités de recherche ou de talents en rédaction. Cette synergie a non seulement enrichi la qualité de notre analyse mais a également offert une plateforme d'apprentissage mutuel, où chacun a pu bénéficier des perspectives et expertises des autres membres. Les différences d'opinion et les débats ont été des éléments inévitables de notre processus de travail, nécessitant une gestion attentive pour s'assurer qu'ils conduisent à des conclusions constructives plutôt qu'à des impasses. La clé de notre réussite a résidé dans l'établissement de règles claires pour la prise de décision et dans l'engagement de tous à respecter les idées de chacun, favorisant ainsi un environnement de travail respectueux et productif. En somme, le travail de groupe sur ce projet a été une expérience précieuse, soulignant l'importance de la communication, du respect mutuel et de la flexibilité. Les défis rencontrés ont été autant d'opportunités d'apprentissage, renforçant notre capacité à travailler efficacement en équipe et à surmonter les obstacles de manière collaborative. Ces leçons resteront inestimables alors que nous progressons dans nos carrières académiques et professionnelles, prêts à affronter les défis futurs avec une approche d'équipe éprouvée et efficace.

14 Conclusion

Après une exploration approfondie des mécanismes sous-jacents à la crise financière de 2008, de ses origines à ses effets persistants sur le système financier global, notre analyse dans le Livrable 2 s'est concentrée sur le calcul de l'Indice du Risque Systémique (IRS), la modélisation de la contagion financière, et l'étude des réseaux financiers. Cette approche méthodique nous a permis de quantifier la vulnérabilité du réseau financier et d'identifier des stratégies potentielles pour renforcer sa résilience face à de futures turbulences.

Le travail réalisé souligne l'importance cruciale d'une surveillance réglementaire adaptative et d'une gestion prudente du risque par les institutions financières. Les modèles dynamiques de contagion financière développés offrent un aperçu précieux de la manière dont les chocs peuvent se propager à travers le réseau, mettant en lumière les points de défaillance critique et les canaux de transmission du risque. De plus, l'analyse des réseaux financiers a révélé des configurations de réseau susceptibles de minimiser la probabilité et l'impact d'une cascade de défaillances.

À travers cette étude, nous réaffirmons que la solidité du système financier repose non seulement sur la santé financière des institutions individuelles mais aussi sur la robustesse de l'ensemble du réseau. La crise de 2008 a démontré que les interconnexions financières, tout en favorisant l'efficience et l'innovation, peuvent également servir de vecteurs pour la propagation rapide du risque systémique. Par conséquent, une approche holistique, intégrant à la fois des mesures microprudentielles et macroprudentielles, est essentielle pour préserver la stabilité financière.

En conclusion, notre exploration dans le Livrable 2 met en évidence l'impératif d'une vigilance continue et d'une adaptation proactive des cadres réglementaires et des pratiques de gestion des risques. Alors que le paysage financier continue d'évoluer, avec l'introduction de nouvelles technologies et la formation de marchés financiers de plus en plus complexes, notre capacité à anticiper et à atténuer le risque systémique sera mise à l'épreuve. Il incombe donc aux régulateurs, aux institutions financières et à la communauté académique de collaborer étroitement pour renforcer les défenses du système financier contre les chocs futurs, assurant ainsi sa résilience et sa durabilité à long terme.