Prenons un exemple concret où nous utilisons la méthode d'inversion pour **simuler le temps entre deux événements** dans un processus de Poisson, modélisé par une loi exponentielle $\text{Exp}(\lambda)$.

Contexte :

Un centre d'appel reçoit en moyenne $\lambda = 2$ appels par minute. Nous souhaitons simuler les durées entre deux appels consécutifs. Ces durées suivent une loi exponentielle avec paramètre $\lambda = 2$.

Étapes :

1. **Fonction de répartition de la loi exponentielle** :

$$F_X(x) = 1 - e^{-\lambda x}, \quad x \geq 0.$$

2. **Inverse de la fonction de répartition** :
   En résolvant $u = 1 - e^{-\lambda x}$ pour x, on obtient :

$$x = -\frac{\ln(1-u)}{\lambda}.$$

- Puisque $U \sim \mathcal{U}(0,1)$, $1 - U \sim \mathcal{U}(0,1)$, donc :

$$x = -\frac{\ln(U)}{\lambda}.$$

3. **Simulation d'une durée entre appels** :
   - Générer $U \sim \mathcal{U}(0,1)$.
   - Calculer $X = -\frac{\ln(U)}{\lambda}$, qui représente la durée simulée.

Implémentation Python :

Voici un code pour simuler les durées entre appels :

```
import random
import math

def simulate_exponential(lambda_):
"""Simule une variable exponentielle avec paramètre lambda."""
U = random.uniform(0, 1)  # Générer U ~ U(0,1)
return -math.log(U) / lambda_

# Paramètre de la loi exponentielle
```

```
lambda_ = 2  # Moyenne : 1/2 minutes entre appels

# Simuler 10 durées entre appels
durations = [simulate_exponential(lambda_) for _ in range(10)]
print("Durées simulées entre appels :", durations)
```

Exemple de résultats :

En exécutant ce script, vous pourriez obtenir des résultats comme ceux-ci (valeurs en minutes) :

```
Durées simulées entre appels : [0.081, 0.344, 0.553, 0.024,
0.137, 0.189, 0.219, 0.017, 0.435, 0.092]
```

Ces durées indiquent les intervalles entre chaque appel. Par exemple :

- Le premier appel arrive après **0.081 minute** (~5 secondes),
- Le suivant après **0.344 minute** (~20 secondes), et ainsi de suite.

Visualisation des résultats :

On peut tracer un histogramme pour observer la distribution des durées simulées :

```
import matplotlib.pyplot as plt

# Tracer un histogramme des durées simulées
plt.hist(durations, bins=10, edgecolor='black', density=True)
plt.title("Simulation des durées entre appels (\u03BB = 2)")
plt.xlabel("Durée (minutes)")
plt.ylabel("Densité")
plt.show()
```

Résultat attendu :

L'histogramme devrait montrer une décroissance exponentielle, conforme à la densité $f_X(x) = \lambda e^{-\lambda x}$.

Application pratique :

Cette simulation est utile pour :

- Modéliser des processus d'attente (comme des files d'attente).
- Étudier le comportement de systèmes soumis à des événements aléatoires (comme des serveurs informatiques).

- Réaliser des études statistiques pour optimiser les ressources nécessaires.