**EXPLICATIONS**

I

*# Estimation de la fonction*

A = np.vander(x, increasing=True)

coefficients = np.linalg.solve(A, y)

y\_gauss = np.polyval(coefficients, x)

lagrange\_poly = lagrange(x, y)

y\_lagrange = lagrange\_poly(x)

newton\_poly = BarycentricInterpolator(x, y)

y\_newton = newton\_poly(x)

Ce code est utilisé pour estimer une fonction à partir d'un ensemble de points de données (x, y) en utilisant différentes méthodes d'interpolation polynomiale. Il crée une matrice de Vandermonde à partir du vecteur x, résout un système linéaire pour obtenir les coefficients du polynôme d'interpolation de Gauss, calcule le polynôme de Lagrange à partir des points de données, et crée un polynôme d'interpolation de Newton basé sur les différences divisées. Ensuite, il évalue ces polynômes aux points x pour obtenir les valeurs estimées y correspondantes. Chaque méthode d'interpolation fournit une estimation de la fonction à partir des données, avec des caractéristiques et des résultats légèrement différents.

II

*# Calcul de l'erreur*

rmse\_gauss = np.sqrt(mean\_squared\_error(y, y\_gauss))

rmse\_lagrange = np.sqrt(mean\_squared\_error(y, y\_lagrange))

rmse\_newton = np.sqrt(mean\_squared\_error(y, y\_newton))

Ce code calcule l'erreur entre les valeurs réelles y et les valeurs estimées y\_gauss, y\_lagrange et y\_newton obtenues à partir de différentes méthodes d'interpolation polynomiale. Il utilise l'erreur quadratique moyenne (RMSE) pour mesurer la différence entre les valeurs réelles et les valeurs estimées. L'erreur est calculée en prenant la racine carrée de la moyenne des carrés des différences entre chaque paire de valeurs réelles et estimées. Le RMSE est une mesure couramment utilisée pour évaluer la précision d'un modèle d'estimation par rapport aux données réelles.

III

*# Visualisation des données*

plt.scatter(x, y, color='red', label='Données originales')

plt.plot(x, y\_gauss, label='Gauss (Directe)')

plt.plot(x, y\_lagrange, label='Lagrange')

plt.plot(x, y\_newton, label='Newton')

plt.xlabel('x')

plt.ylabel('y')

plt.legend()

plt.show()

Ce code utilise la bibliothèque matplotlib pour afficher un graphique qui représente les données originales (x, y) ainsi que les fonctions estimées (y\_gauss, y\_lagrange, y\_newton) à partir de différentes méthodes d'interpolation. Les données originales sont affichées sous forme d'un nuage de points rouges, tandis que les fonctions estimées sont représentées par des courbes. L'axe des abscisses représente la variable x, l'axe des ordonnées représente la variable y, et une légende est ajoutée pour identifier chaque courbe. En visualisant ce graphique, il est possible de comparer les performances des différentes méthodes d'interpolation et de les évaluer par rapport aux données réelles.