Projet « une aire » de déjà-vu

Soit f une fonction continue sur `[a,b]`. Pour ce projet, il s'agira d'approcher le calcul de l'intégrale :

$$\int_a^b f(x)dx$$

1. méthode des rectangles

Dans la méthode dite des rectangles (interpolation d'un polynôme de degré 0), on a le choix entre trois techniques :

1. on fait coïncider le sommet haut gauche du rectangle avec la courbe : méthode des rectangles à gauche,
2. on fait coïncider le sommet haut droit du rectangle avec la courbe : méthode des rectangles à droite,
3. on fait coïncider le milieu du côté haut du rectangle avec la courbe : méthode du point milieu

On définit le pas d'approximation :

$h = \frac{b-a}{n}$ où `n` est le nombre de rectangles avec lesquels nous allons « *paver* » l'aire à calculer.

On définit ainsi : $x_i = a + ih$

Avec la méthode du point milieu, l'aire se calcule de la manière suivante :

$$\int_a^b f(x)dx \approx \sum_{i=1}^{n-1} (x_{i+1} - x_i) f(\frac{x_i + x_{i+1}}{2})$$

Ecrire trois fonctions GNU Octave :

- **`integration_rectangles_D`** qui utilise la méthode des rectangles à droite.
- **`integration_rectangles_G`** qui utilise la méthode des rectangles à gauche.
- **`integration_rectangles_M`** qui utilise la méthode des rectangles du point milieu.

2. méthode des trapèzes

La méthode des trapèzes (interpolation d'un polynôme de degré 1) est à peu près similaire à la méthode précédente mais on utilise maintenant des trapèzes pour « *paver* » l'aire afin d'être plus précis.

Comme plus haut, l'intervalle `[a,b]` est partagé en n petits trapèzes de largeur `h = (b-a)/n`. Nous savons ainsi que l'aire de chaque petit trapèze est `Ai = (h/2)*(f(a+ih) + f(a+(i-1)h))`.

Ecrire une fonction GNU Octave **`integration_trapezes`** qui utilise la méthode des trapèzes.

3. méthode de Simpson

Cette méthode utilise l'approximation de f par un polynôme quadratique (de degré 2) prenant les mêmes valeurs que f aux points d'abscisse a, b et m =(a+b)/2.

$$\int_a^b f(x)dx \approx \frac{b-a}{6}\left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right)$$

Par ailleurs, plus l'intervalle est petit, meilleure est l'approximation de la valeur de l'intégrale. Par conséquent, pour obtenir un résultat correct, on subdivise l'intervalle [a,b] en un nombre pair de sous-intervalles et on additionne la valeur obtenue sur chaque sous-intervalle (méthode de Simpson *composite*)

Ecrire une fonction GNU Octave **`integration_simpson_composite`** qui utilise la méthode de Simpson composite.

4. méthode de Monte-Carlo

Le terme de « *méthode de Monte-Carlo* » désigne toute méthode visant à calculer une valeur numérique en utilisant des procédés aléatoires, c'est-à-dire en utilisant des techniques probabilistes. Le nom de ces méthodes, qui fait allusion aux jeux de hasard pratiqués à Monte-Carlo, a été proposé par l'équipe du physicien Nicholas Metropolis.

Soit X une variable aléatoire réelle ayant une densité de probabilité uniforme sur l'intervalle [a,b]. Soit **n** échantillons x_i de cette variable aléatoire. L'évaluation de l'intégrale par la méthode Monte-Carlo consiste dans sa forme élémentaire à calculer la somme suivante

$$\int_a^b f(x)dx \approx \frac{b-a}{n}\sum_{i=0}^{n-1} f(x_i)$$

Ecrire une fonction GNU Octave **`integration_MC`** qui utilise la méthode de Monte-Carlo.

Affichage

Pour chacune des fonctions, écrire un **script** permettant de choisir la fonction à afficher dans intervalle choisi et la précision obtenue en fonction du nombre d'intervalles.

Evaluation

Durant ce projet, vous utiliserez 5 fonctions qui vous serviront d'illustration pour vos calculs (par défaut, vous utiliserez les bornes suivantes [-5,5]) :

$$f(x) = \cos(\mathrm{x}^3)$$

$$g(x) = \mathrm{x}\sin(\frac{1}{x}\sqrt{|1-x|}\,)$$

$$h(x) = \ln\,(|\sin(\mathrm{x})|)$$

$$i(x) = \frac{x}{1+x^4}$$

$$j(x) = \frac{4}{3}x sin(e^{\mathrm{x}})$$

Au-delà de l'aspect graphique, il est indispensable de **<u>quantifier</u>** la qualité des calculs effectués.

La qualité va s'évaluer quand cela est possible en comparant les résultats numériques obtenus avec <u>une</u> valeur de référence (*l'intégrale calculée analytiquement par exemple*)

Les méthodes de calcul seront évaluées *a minima* au moyen de trois variables : le **nombre d'itérations** utilisées pour effectuer pour le calcul, **l'erreur par rapport à la valeur de référence** et le **temps de calcul** de l'intégration pour les fonctions proposées.

Les résultats d'évaluation seront présentés :

1. sous la forme d'une matrice pour chaque fonction où pour chacune des méthodes utilisée, nous trouverons pour chacun des intervalles demandés : le résultat de référence de l'intégrale, le résultat trouvé avec la méthode utilisée, la précision demandée et le temps de calcul.
2. sous la forme de graphiques représentant pour chaque méthode, le nombre d'itérations utilisées par rapport à la précision demandée.

Conclure en argumentant sur la meilleure méthode à utiliser pour calculer une intégrale.

Travail à rendre

Vous devrez rendre pour le **vendredi 13 janvier 2023 23h55 GMT dernier délai** par voie électronique (à l'adresse : **Clement.Truillet@irit.fr** ou **Philippe.Truillet@irit.fr** suivant votre groupe de TP) :

- Les fichiers **.m** que vous aurez créés. Il vous est conseillé de faire un fichier pour chaque fonction de calcul un fichier pour l'évaluation et la comparaison des méthodes de calcul ;
- Un rapport décrivant les algorithmes que vous aurez programmés en expliquant votre démarche de travail.

Une présentation orale de 20 minutes (avec documents pour la vidéo-projection) aura enfin lieu après la fin du projet **le 18 janvier 2023.**