

## Consignes :

- Ce travail peut se réaliser seul ou à deux. Dans le cas d'une production à deux, la réalisation doit être partagée de manière relativement égale entre les deux participants. Au cas où ce ne serait pas le cas, la note individuelle reflètera cette différence. L'enseignant peut demander à votre historique de commit à cet effet.
- Toute utilisation d'une IA doit être ciblée et ponctuelle, faire l'objet d'une citation claire, et être accompagnée d'une compréhension détaillée de chaque instruction

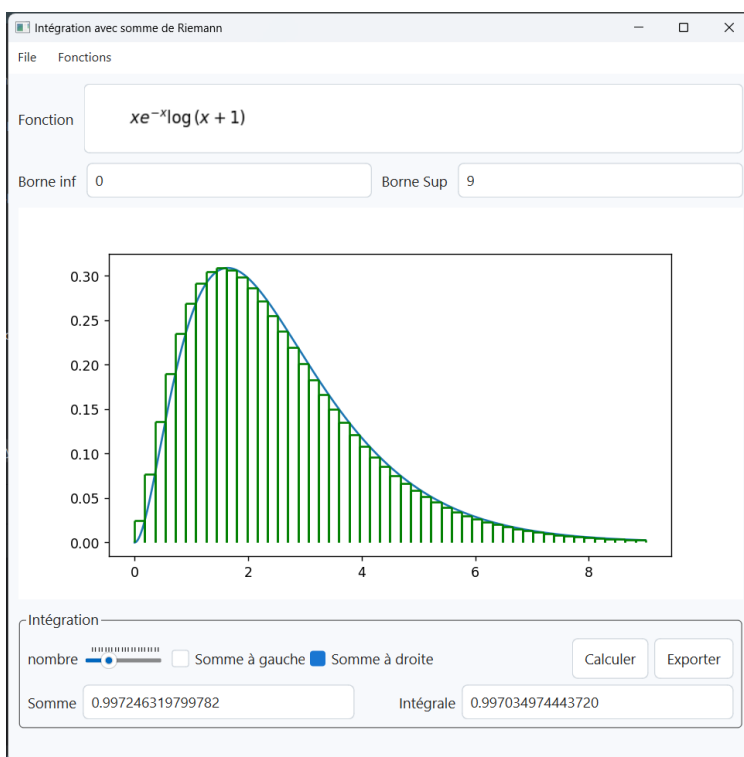
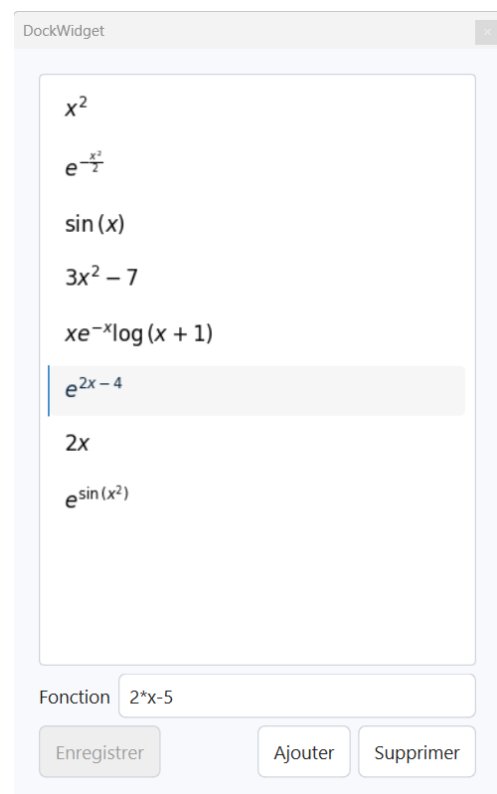
## Remise :

- Le projet complet sur LEA (sans son venv trop volumineux). Vous devez conserver un historique de commit précis et soigné, qui sera utilisé au besoin le volume et la qualité du travail effectué par chacun.

**Date :** lundi 27 octobre à minuit

**Objectif :** Réaliser une application permettant d'afficher le graphe d'une fonction entre 2 bornes choisies par l'utilisateur puis de calculer une somme de Riemann (à gauche ou à droite) afin d'approcher la valeur de l'intégrale. On souhaite afficher les rectangles permettant de calculer la somme. On veut également pouvoir éditer la liste des fonctions qui peuvent être affichées et la sauvegarder dans un fichier json

Voici une image indicative de cette application.

Vous trouverez ci joint une vidéo détaillant les fonctionnalités que vous devez implémenter

## Architecture :

Votre programme doit contenir au moins 5 classes différentes :

- La fenêtre principale (Vue)
- La fenêtre de la liste des fonctions (Vue)
- Le Canvas Matplotlib pour afficher les graphes et les sommes (Vue)
- La classe qui maintient la liste des fonctions (Modèle)
- La classe qui maintient la fonction à intégrer et les propriétés de la somme à calculer (bornes, gauche ou droite, valeur de l'intégrale, de la somme, etc..) (Modèle)

Il est important de **respecter les responsabilités de chacune**. En particulier on vous demande autant que possible de séparer les vues des modèles. Aucun traitement lié à l'affichage ne doit se trouver dans les modèles, alors que la logique et les mathématiques doivent se trouver uniquement dans les modèles.

## Étapes suggérées :

- 1) Créer la fenêtre principale avec un champ (`QLineEdit`) pour la fonction et un pour chaque borne.
- 2) Créer le canvas, et implémenter le dessin du graphe. Lorsqu'on modifie la fonction ou une des bornes, la fonction doit se redessiner en tenant compte des nouvelles valeurs.
- 3) Ajouter le group box qui gère la somme de Riemann, implémenter **la méthode du modèle** qui calcule la somme de Riemann et celle qui calcule l'intégrale. Gérer les états des boutons et des champs en fonction du contexte.
- 4) Implémenter l'affichage des rectangles qui constituent la somme dans le canvas matplotlib
- 5) Ajouter une vue qui va permettre d'éditer une liste de fonctions (utiliser une `QListView`) avec un bouton ajouter et un bouton supprimer. Porter une attention particulière à la séparation vue modèle.

- 6) Implémenter la sauvegarde de la liste en json, puis le chargement de la liste depuis le json au démarrage de l'application.
- 7) Dans la vue principale remplacer le champ d'édition de la fonction par un ComboBox qui affiche au démarrage la liste des fonctions chargées dans le modèle à partir du json.
- 8) Implémenter un mécanisme de signal personnalisé qui permet au Combo Box de se mettre à jour chaque fois que la liste de fonctions est modifiée.
- 9) Tester soigneusement l'interface et les différentes synchronisations
- 10) Effectuez des recherches pour afficher les fonctions de manière plus élégante, en utilisant LaTeX. Je vous suggère ici de vous faire aider par un outil IA de votre choix qui va vous suggérer des solutions que vous devrez vous approprier.
- 11) Implémenter ensuite les fonctionnalités résiduelles :
  - Validation en temps réel sur les bornes
  - Validation complète de la fonction à l'ajout
  - Redimensionnement de fenêtres cohérent
  - Export de l'image
  - Possibilité de Docker la fenêtre d'édition de la liste de fonctions
  - Appliquer un css de votre choix
  - Toute autre bonne idée est bienvenue!