



CY Cergy Paris Université

**CY Tech – Grande école de sciences, d'ingénierie,
d'économie et de gestion**

Projet WildWater

Par :

Sarah ALLOUI, Fatumata JALLOH , Maria TAZI

(Filière : Pré-ingénieur Mathématiques Informatique Option Physique)

Groupe MI3-F

Sous l'encadrement de : M GALOUZ Zinedine

Année universitaire : 2025/2026

Sommaire

I. Organisation du Travail	3
II. Description des Fonctionnalités	4
III. Problèmes rencontrés	5
Conclusion	5

I. Organisation du Travail

Dans le cadre de notre projet, nous avons réparti les tâches entre les trois membres de l'équipe afin d'assurer une organisation efficace et une bonne gestion du temps.

Chaque membre s'est vu attribuer une partie précise du projet, en fonction des compétences de chacun et des besoins techniques.

Sarah était responsable de la partie Histogrammes. Elle a travaillé sur la lecture du fichier CSV, le stockage des usines dans un arbre AVL, le calcul des volumes (*max*, *src*, *real*) et la génération des fichiers CSV destinés à la création des histogrammes.

Fatumata s'est occupée de la construction de l'arbre représentant le réseau de distribution d'eau. Elle a mis en place les structures de données nécessaires pour représenter les différents nœuds (usines, jonctions, stockages, usagers), ainsi que le calcul des pertes d'eau et du rendement.

Maria a pris en charge l'intégration globale du projet. Elle a développé le script Shell servant de point d'entrée, permettant de vérifier les arguments, de lancer automatiquement la compilation, d'exécuter le programme C et de mesurer le temps d'exécution.

II. Description des Fonctionnalités

histogrammes (Sarah)

Cette partie du projet permet de représenter les volumes d'eau par usine sous forme d'histogrammes. Le programme lit le fichier CSV et stocke chaque usine dans un arbre AVL à l'aide de son identifiant.

Pour chaque usine, trois valeurs sont prises en compte :

- la capacité maximale (*max*),
- le volume capté (*src*),
- le volume réellement traité (*real*).

Selon le type d'histogramme demandé, un fichier CSV de sortie est généré. Les usines sont affichées par ordre alphabétique inverse grâce à un parcours spécifique de l'arbre AVL. Ces fichiers sont ensuite utilisés par **gnuplot** pour produire les histogrammes.

Les fuites et le rendement (Fatumata)

Cette partie consiste à analyser les pertes d'eau dans le réseau de distribution. Le programme construit une structure représentant le réseau, où chaque élément correspond à un point du système (usine, jonction, stockage ou usager).

À partir d'une usine donnée, le réseau est parcouru afin de calculer la quantité d'eau perdue à cause des fuites. Les données sont traitées de manière rigoureuse afin d'éviter les erreurs et d'obtenir des résultats fiables.

Les résultats sont ensuite enregistrés dans un fichier pour conserver une trace des calculs effectués.

Le script Shell et l'intégration (Maria)

Le script Shell permet de lancer le programme de manière simple et efficace. Il vérifie la validité des paramètres fournis et s'assure que le programme est bien compilé.

Si nécessaire, le script déclenche automatiquement la compilation, puis exécute le programme avec les arguments appropriés. Il gère également les fichiers générés et affiche le temps d'exécution, facilitant ainsi l'utilisation globale du projet.

III. Limitations rencontrées

- **Temps d'exécution** : le traitement d'un fichier CSV de 500 Mo est relativement lent (environ 5 minutes). L'application n'est pas optimisée pour des volumes de données très importants.
- **Affichage Gnuplot** : les noms des usines étant parfois trop longs, ils peuvent se chevaucher sur les histogrammes, ce qui réduit la lisibilité.
- **Gestion des arguments** : l'utilisation de guillemets est obligatoire pour les identifiants contenant des espaces, sous peine d'erreur lors de la recherche.

Conclusion

Le projet **WildWater** a été une expérience enrichissante qui nous a permis de mieux comprendre le fonctionnement d'un projet informatique complet. Nous avons appris à lire et traiter des fichiers CSV, à structurer les données à l'aide d'arbres AVL et à produire des résultats exploitables sous forme de fichiers de sortie.

Ce projet nous a également permis de progresser en langage C et en Shell, notamment pour automatiser la compilation, gérer les arguments et simplifier l'exécution du programme. Enfin, ce travail en groupe nous a appris à mieux nous organiser, à répartir efficacement les tâches et à comprendre l'importance d'un code structuré, testé et fiable.

