

RAPPORT DE PROJET -CyTech Pau

PROJET “CYMeteo” :
*Application de
traitement
météorologique*

*Projet réalisé par la “Team T.A.B”
Théo DE MORAIS,
Ayman EL KILI,
Baptiste PLAUT-AUBRY,
en Pré Ing2 Groupe 1 à CyTech*

Projet encadré par
Thierry GARCIA
Ines DE COURCHELLE
Elisabeth RANISAVLEJVIC

Lien du projet GitLab CY : <https://gitlab.etude.cy-tech.fr/team-tab-info-projet/projet-cy-meteo>

Année 2022-2023

Sommaire

1)Organisation et Répartition des tâches au sein du groupe

2)Planning de réalisation

3)Limitations fonctionnelles

4)Résultats obtenus

5)Sitographie

1) Organisation et Répartition des tâches au sein du groupe :

Dans un premier temps nous avons effectué un travail de recherches sur les différents points du sujet fourni :

Il a fallu lire et comprendre les consignes générales et s'adapter aux libertés laissées par le sujet. Il nous a fallu un certain temps avant de comprendre comment allait s'organiser le squelette du programme afin de se répartir au mieux les tâches.

Il a fallu analyser le tableau CSV fourni et faire des recherches sur les différentes stations météorologiques avec les sources à notre disposition.

Nous avons donc d'abord répertorié les options du Shell dans un tableau afin de déterminer les IDs exclusives à chacun et avons commencé nos recherches sur le sujet, les fonctions et l'organisation que nous allions adopter. Les recherches sur Gnuplot ont eu lieu plus tard, car bien qu'étant essentiel dans le résultat final du programme, il ne sert à rien dans les premiers pas de la programmation.

Dans un premier temps, nous nous sommes répartis les tâches comme suit:

- Théo :

Comme, lors du projet de l'année passée, je me suis occupé de la fonction principale, nous avons décidé que je m'occuperai d'abord de cette même fonction lors ce projet.

J'ai ainsi pu traiter, à mon niveau, les différentes options pouvant être indiquées dans le terminal par l'utilisateur de la manière la plus efficace possible. C'est une fois les bases posées au sein de cette fonction qui a été bien évidemment amenées à de multiples modifications par la suite, que j'ai pu m'atteler à la suite du projet. Et plus précisément, aux différentes fonctions qu'allait utiliser le programme C. Différentes fonctions d'abord créées par Baptiste concernant les arbres n'ont malheureusement pas abouties. Effectivement, nous voulions, dans un premier temps, traiter le projet avec l'aide des arbres. Cependant, nous nous sommes vite rendus compte que l'utilisation des tableaux était plus simple. C'est alors que j'ai pu effectuer les différentes fonctions de tris dans le programme C.

- Ayman :

En lisant le sujet et en le comprenant bien, nous nous sommes réparti les tâches. Pour ma partie de travail je me suis concentré sur GNUPLOT et son mode de fonctionnement. En premier lieu, il s'agissait d'idées éparées et peu claires, ce qui m'a incité à rechercher le plus d'informations possible, ce qui m'a fait découvrir que GNUPLOT est très important pour tracer n'importe quel type de diagramme. Après des heures de recherche continue, les choses ont commencé à devenir plus claires. Après avoir bien compris le fonctionnement de GNUPLOT et ce qui prend en entrée et sortie, j'ai commencé à rédiger des codes en script shell et en les testant parallèlement avec un fichier de test pour obtenir finalement le diagramme demandé dans le sujet, et c'était le cas pour les différents types de diagrammes demandés dans le sujet du projet.

- **Baptiste :**

Intéressé par le fonctionnement du shell que l'on avait commencé à aborder en cours quelques semaines plus tôt, j'ai attaqué cette partie-là en premier. En y allant étapes par étapes avec ce qui était demandé j'ai d'abord géré les arguments géographiques exclusifs puis avec plus de difficulté, les arguments liés à la récupération de la date min et max, l'affichage du temps d'exécution...

Après que les choses aient bien avancé j'ai construit les fonctions relatives au tri des données d'un arbre ABR grâce à ce que l'on avait déjà fait au cours de l'année dans les TDs j'ai essayé de réaliser des tris dans le tableau en utilisant ces fonctions de manière plus ou moins convaincante, avec plusieurs difficultés. Finalement Théo s'est attaqué au traitement des données dans le c avec des arbres. Je me suis aussi occupé de vérifier les fichiers C et exécutif C en mettant en lien avec le Shell le programme C, passant des arguments au Shell.

Grâce aux recherches de Ayman sur Gnuplot, j'ai découvert qu'il était possible de réaliser la plupart des tris avec des commandes shells simples, comme awk, grep et sort. Pour la fin du projet, j'ai donc mis en place ces tris tout en coordonnant les données des tableaux avec les diagrammes Gnuplot.

Il faut noter qu'il a été difficile de traiter le fichier `meteo_filtered_data_v1.csv` en raison de sa taille importante (250 mo). Cela a rendu son traitement difficile sur les ordinateurs de l'école et sur les machines virtuelles/petits serveurs. Nous avons donc dû utiliser des échantillons de ce fichier, ce qui a pu fausser certains des choix en termes de complexité. Cependant, cela m'a (je l'espère) appris à gérer des difficultés liées à la complexité et m'obligera à l'avenir à me pencher sur des solutions plus efficaces lors de réalisations de projets.

2)Planning de réalisation :

- **Théo :**

Semaine 05/12 :

Début du projet, mise en place de l'organisation, création du gitlab et des dépôts. Rassemblement des infos essentielles sur le projet, compréhension du sujet et premières lignes de codes.

Semaine du 19/12 :

Après la fin des cours et des derniers examens, début du travail concret. Réalisation majeure de la fonction principale du programme.

Semaine du 26/12 :

Fonction principale du programme en partie terminée, modifications à venir en fonction de l'avancement sur les différentes fonctions de tri effectuées. Réflexions et tests sur la lecture du fichier et les manières de trier ce dernier.

Semaine du 02/01 :

Tests avec quelques fonctions de tri en utilisant les arbres. Cela s'est avéré plus compliqué que prévu.

Semaine du 09/01:

Sans succès avec les arbres, je me suis penché sur les tableaux. J'ai ainsi pu effectuer la lecture du fichier pour chaque fonction de tri et réaliser l'écriture des résultats désirés dans un fichier de sortie. Les fonctions créées et opérationnelles dans le programme C sont les suivantes : humidité, hauteur, température mode 1 et pression mode 1. La persévérance a enfin porté ses fruits !

Semaine du 16/01

Lors de l'écriture des résultats finaux dans le fichier de sortie, nous nous sommes rendus compte que les identifiants et les dates de stations n'affichent pas les 0. Par exemple, pour les identifiants, au lieu d'afficher 07105, il y avait 7105. De même pour les dates, il y avait 2010-1-5 au lieu de 2010-01-05. J'ai ainsi réglé le problème pour chacune des fonctions.

Enfin, j'ai rédigé les commentaires du programme C de façon à expliquer chacune des actions effectuées au sein de ce dernier.

- **Ayman :**

Semaine 05/12 :

Début du projet, mise en place de l'organisation, création du gitlab et des dépôts. Rassemblement des infos essentielles sur le projet, compréhension du sujet et premières lignes de codes.

Semaine du 19/12 :

J'ai commencé à faire des recherches sur GNUPLOT et comment ça fonctionnait.

Semaine du 26/12 :

J'ai essayé de comprendre les différents types de diagrammes, et ce que ça prenait en entrée .

Semaine du 02/01 :

j'ai rédigé les codes script shell pour chaque diagramme en les testant parallèlement pour vérifier si tout marche tranquillement.

Semaine du 09/01:

j'ai finalisé les programmes et essayé de les adapter avec les fichiers d'entrées.

- **Baptiste :**

Semaine 05/12 :

Début du projet, mise en place de l'organisation, création du gitlab et des dépôts. Rassemblement des infos essentielles sur le projet, compréhension du sujet et premières lignes de codes.

Semaine du 19/12 :

Après la fin des cours et des derniers devoirs, début du travail concret, réalisation de l'affichage exclusif, et tests multiples.

Semaine du 26/12 :

Fonctions ABR, Vérifications du C, temps d'exécution des programmes.

Semaine du 02/01 :

Prises des arguments du C, avancées sur gnuplot, stoppées par une grosse grippe (rip), nous avons commencé à travailler sur les tris en utilisant des arbres mais cela s'est avéré plus compliqué que prévu.

Semaine du 09/01:

Pendant que nous avions du mal à faire progresser le projet avec des arbres malgré les efforts de Théo, nous avons décidé et remarqué qu'il serait finalement bien plus simple de réaliser les tris avec le script shell, qui pouvait ensuite être utilisé facilement dans gnuplot. Nous avons donc concentré nos efforts sur cet objectif. Le 13 nous avons réalisé la plupart des diagrammes une fois avec succès (sauf vent et plus ou moins mode 3,2 avec difficultés) et en fin de semaine le projet commençait à entrer dans sa forme finale.

Semaine du 16/01 :

Réglages mode 2,3 ajout -a -g, réalisation du vent, modifications diverses pour diminuer la complexité du programme et finitions du projet.

3) Limitations fonctionnelles du projet :

-Pour des raisons pratiques, nous avons mis le -a -g dans le script shell plutôt que dans le C, le -d aussi mais cela est lié au départ à un flou dans la consigne où il est spécifié d'avoir le -d dans le c et dans le shell.

-Il faut **obligatoirement**, dans le shell en cas d'utilisation d'une option géographique la rentrer en premier lors de l'exécution du script shell. Il aurait été possible de faire autrement mais l'investissement aurait rendu le code encore plus gros pour un résultat minime.

-Par souci de temps liés à plusieurs facteurs (partiels, Noël indisponible etc), nous avons (ici surtout moi Baptiste en particulier) décidé de réaliser plusieurs tris en shell plutôt qu'en C.

Si au départ cela ne semblait avoir uniquement des avantages car bien plus rapides à coder, nous nous sommes heurtés au problème de temps de complexité. Si toutes les options fonctionnent, elles restent néanmoins plus lentes à exécuter que des tris C classiques.

Temps de Complexité sur le meteo filtered sans filtre : Testé sur le PC de Théo sur

-H : 8 sec

-M : 8 sec

-V : 58 sec

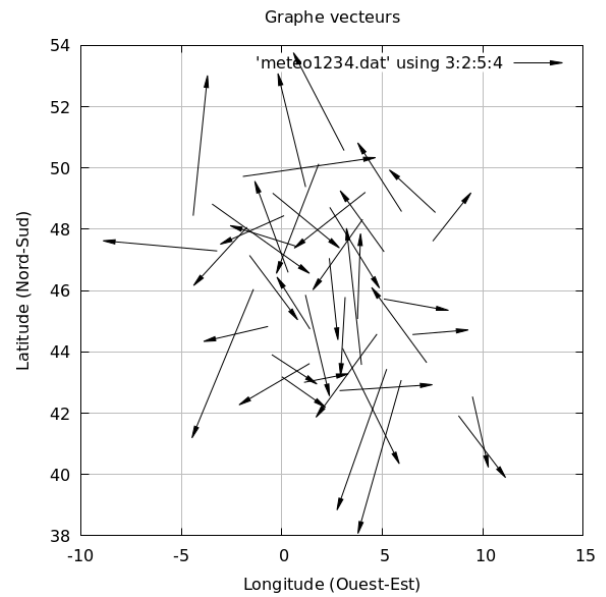
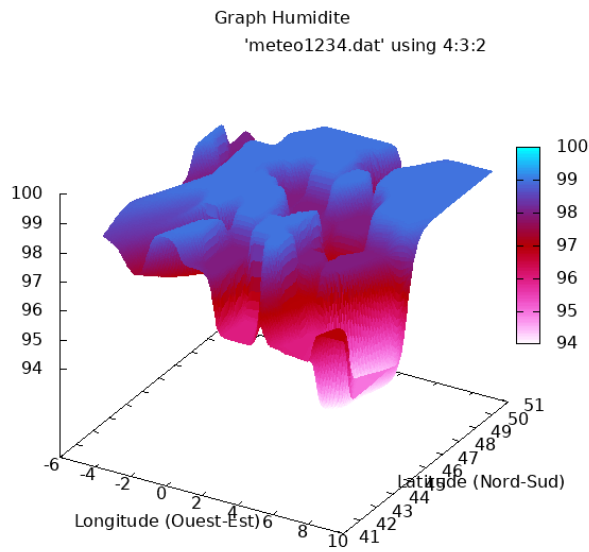
-T 1 : 30 secondes

-T 2 : 10 secondes

-T 3 : 9 secondes

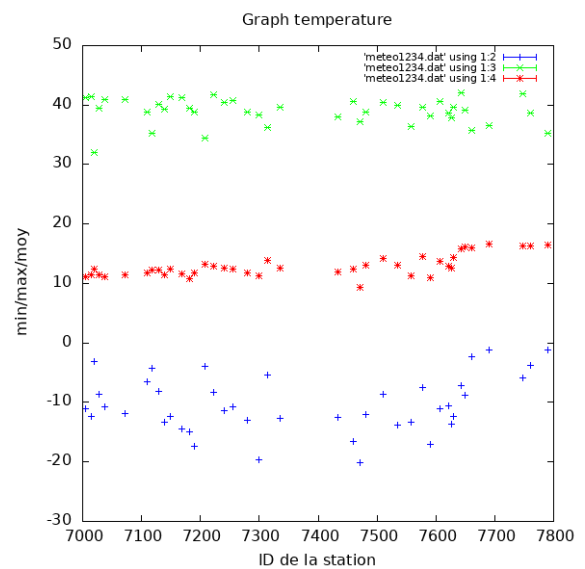
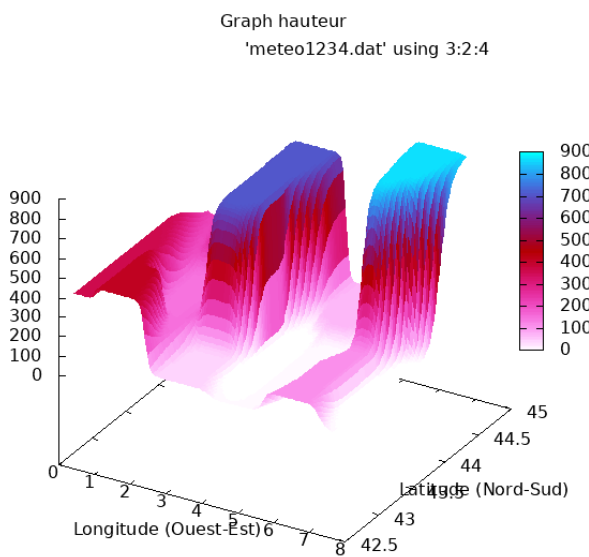
PS :Baptiste : Je pensais pouvoir obtenir des temps relativement proches d'un traitement avec des tris en C mais j'ai dû admettre que la puissance de calcul entre un langage compilé comme le C et un langage interprété par le Shell était bien plus importante que ce que je pensais. S' il était trop tard pour tout changer, j'aurais néanmoins appris l'importance de choisir un langage de programmation adapté aux enjeux et aux contraintes (en l'occurrence l'utilisation de la mémoire vive et d'un fichier contenant des quantités massives de données). Fort heureusement Théo à pu de son côté coder quelques fonctions en C, prouvant la supériorité en termes de temps de calcul des 2 méthodes.

4) Quelques Résultats obtenus



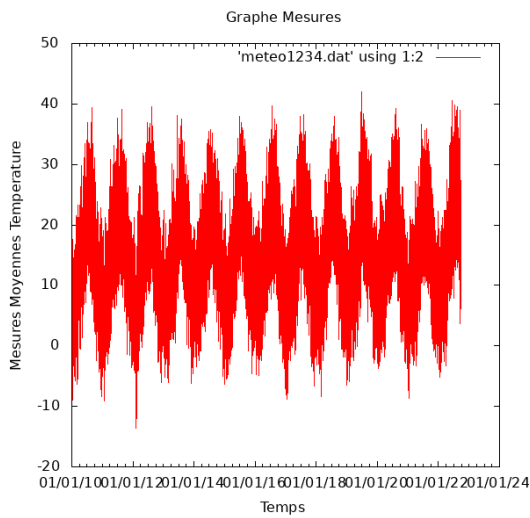
Humidité France :

Vents France :

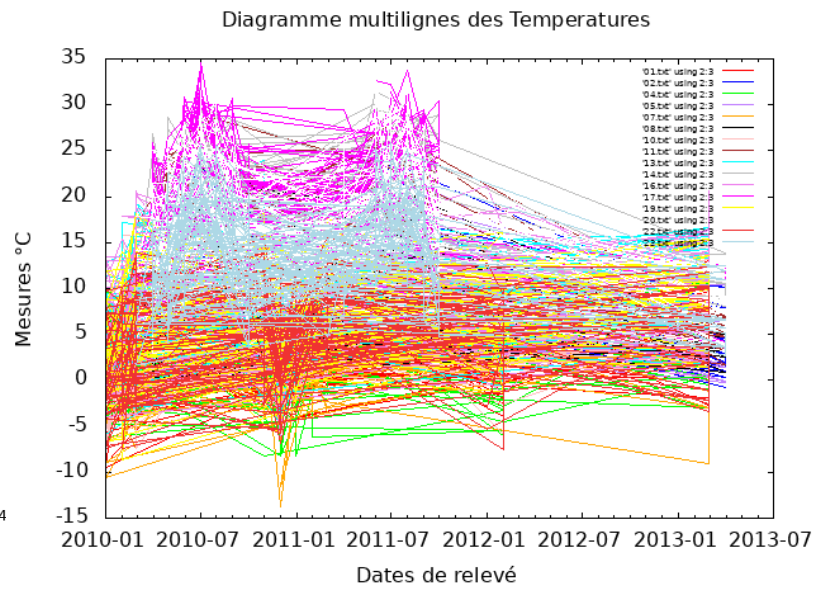


-Hauteur France + Lat 0 45 + Long 0 8 :

-T°C Mode 1 Fr :



T°C Mode 2 FR



T°C Mode 3 FR

Note : Des résultats équivalents mais moins “esthétiques” ont été obtenus pour les modes de pression. et tous les modes au global

5) Sitographie

Voilà pour conclure quelques sources utilisées pour réaliser le projet :

Librairie C en général, clock et ouverture fermeture de fichiers :

<https://koor.fr/C/Index.wp>

http://wiki.juneday.se/mediawiki/index.php/Chapter:C_memory_tools

Tris de données par arbres ABR :

<https://www.geeksforgeeks.org/c-programming-language>

<http://ide.perso.eisti.fr/cours/INFO3/CM/cm5/cm5.html#1>

Commandes et Manuels d'utilisation en script Shell :

<https://tldr.sh/>

https://www.linuxcommand.org/lc3_adv_awk.php

Gnuplot :

<https://livebook.manning.com/book/gnuplot-in-action-second-edition/table-of-contents/>

<http://www.gnuplotting.org/plotting-data/>

<https://gnuplot.sourceforge.net/demo/>

https://gnuplot.sourceforge.net/docs_4.2/node259.html

Le shell et les commandes en général :

<http://ide.perso.eisti.fr/cours/INFO3/CM/cm7/cm7.html>

<http://ide.perso.eisti.fr/cours/INFO3/CM/cm8/cm8.html>

(L'incroyable) fonction awk :

https://linuxhint.com/conditional_statement_awk_command/

<https://archive.org/details/pdfy-MgN0H1joIoDVoIC7/page/n73/mode/2up>

<https://blog.robertelder.org/intro-to-awk-command/>

<https://www.grymoire.com/Unix/Awk.html>

-----Nous Contacter-----

Baptiste Plaut-Aubry

Ayman El Kili

Theo De Morais

plautaubry@cy-tech.fr

elkiliayma@cy-tech.fr

demoraisth@cy-tech.fr