Appel de fonctions Python / Matlab à partir de LabVIEW

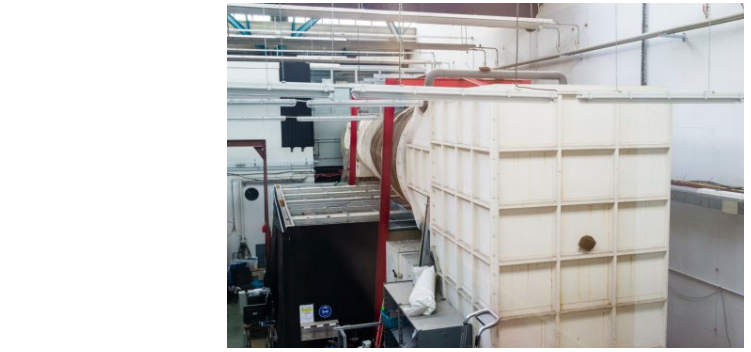
Un exemple avec la création de fichier NetCDF

Muriel Lagauzère, Théo Fernandez

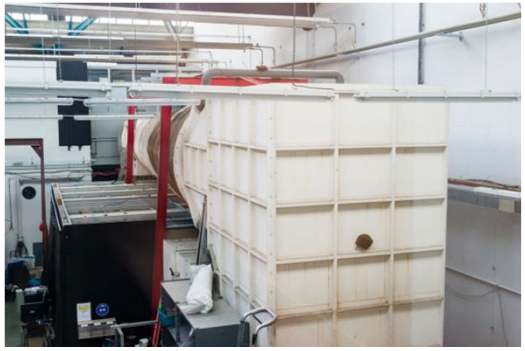
Laboratoire des Écoulements Géophysiques et Industriels (LEGI), UMR 5519, Grenoble, France

Rencontre AlpesVIEW/CNRS, Jeudi 14 Novembre, 2024, Institut Néel, Grenoble, France

Sommaire

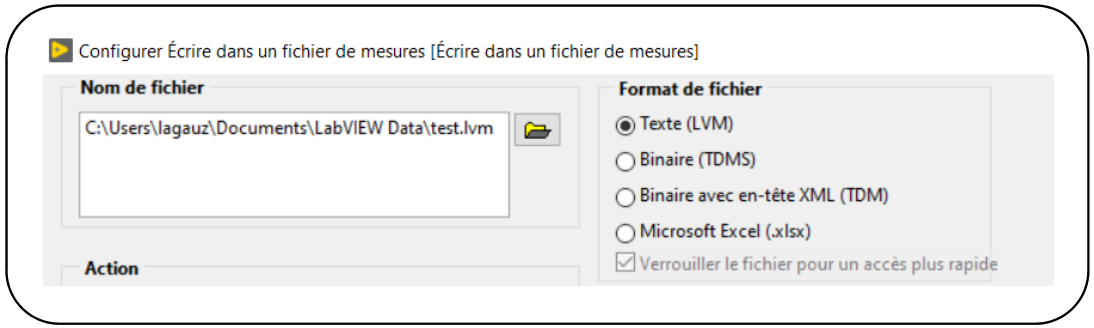


Sommaire



Codes d'acquisition et d'enregistrement des données

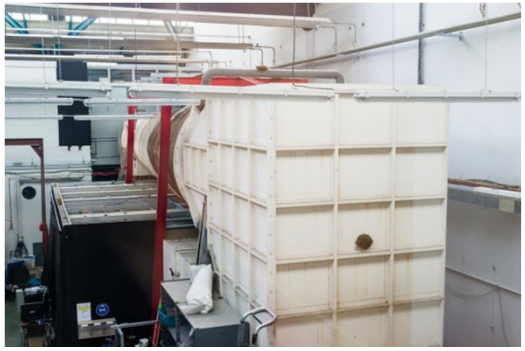
- Plusieurs programmes d'acquisition sur LabView sur différentes expériences
- Différents formats d'enregistrement des données (tableur, texte, lvm, binaire…) avec ou sans entête



Volonté d'obtenir des fichiers mieux documentés, en accès libre pour un échange de données scientifiques

NetCDF (Network Common Data Form)

Sommaire

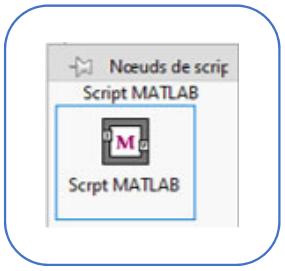


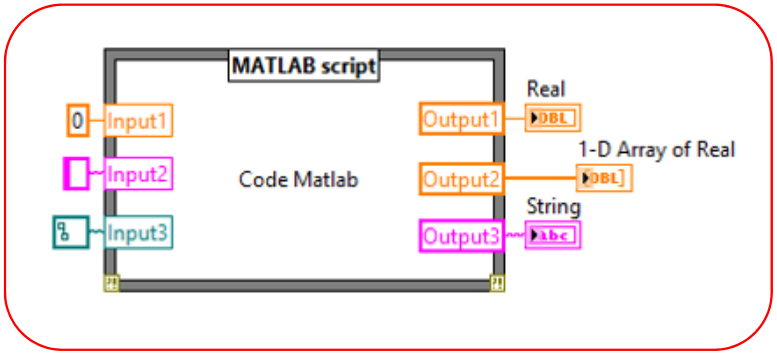
Nœud de script Matlab

Prérequis :

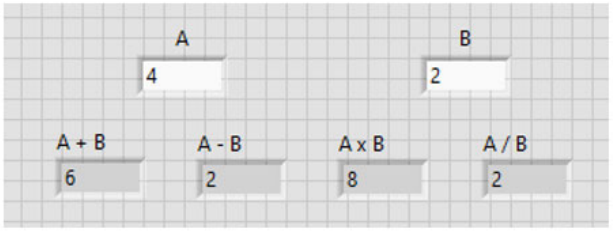
- Avoir le logiciel **Matlab**, version 6.5 ou supérieure, installé sur son ordinateur.

Matlab est un « *Nœud de script* » et fonctionne comme une « *Boîte de calcul* »

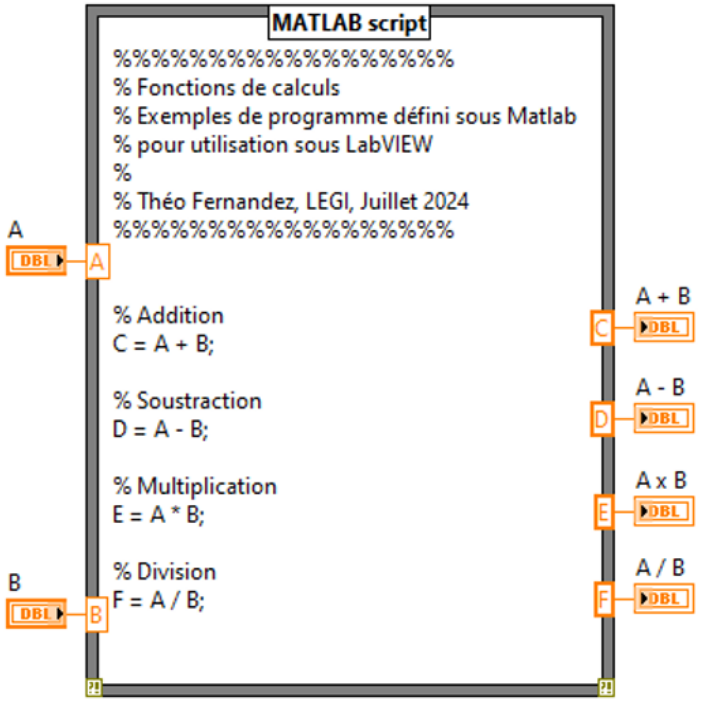




Exemple d'intégration sur LabVIEW



Face-avant



Diagramme

Prérequis pour utilisation de Python

- Si Labview est en 64 bits, choisir une version Python 64 bits (idem pour 32 bits)
- Installer la version de Python compatible avec votre version de LabVIEW

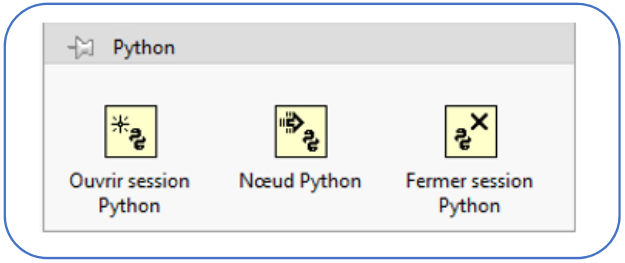
LabVIEW Version	Python Version					
	3.10	3.9	3.8	3.7	3.6	2.7
2024 Q3	■	■	■	■	■	
2024 Q1	■	■	■	■	■	
2023 Q3	■	■	■	■	■	
2023 Q1	■	■	■	■	■	
2022 Q3		■	■	■	■	
2021 SP1		■	■	■	■	
2021		■	■	■	■	
2020 SP1					■	■
2020					■	■
2019 SP1					■	■
2019					■	■
2018 SP1					■	■
2018					■	■

Source: *ni.com*

- Choisir l'installation personnalisée pour que Python s'installe dans le répertoire C:/ProgramFiles/ et soit accessible par tous les utilisateurs.

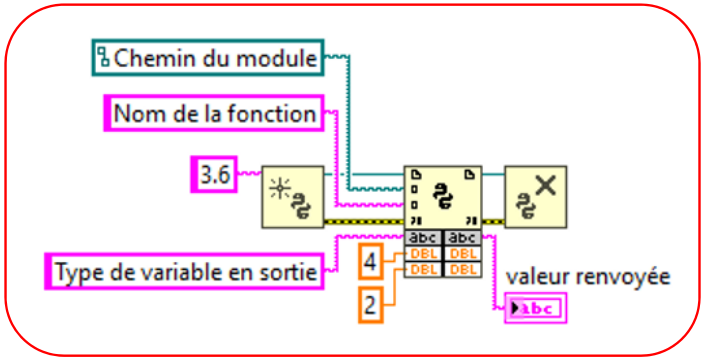
Appeler du code Python à partir de LabVIEW

Python se présente sous forme d'un « *Nœud Python* », qui fait appel directement à une **fonction** Python

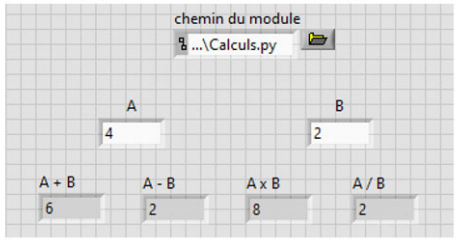


Utilisation:

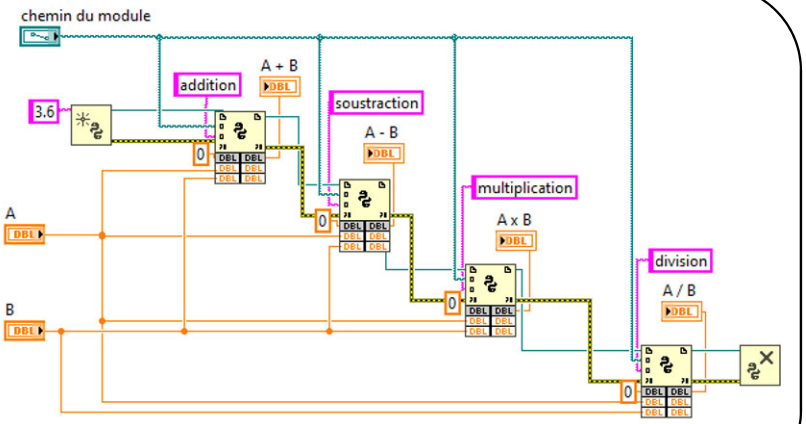
- **Ouvrir une session Python** (ajouter la version de Python)
- Utiliser un **nœud python** (appel du chemin du module, du nom de la fonction, variables d'entrées et de sortie, type de sortie). **Attention, les variables d'entrées doivent être câblées dans le même ordre que les paramètres d'entrées de la fonction Python**
- **Fermer la session python**



Exemple d'intégration sur LabVIEW



Face-avant



Diagramme

LabVIEW

```python
# -*- coding: utf-8 -*-
"""
Fonctions de calculs
Exemples de fonctions définies sous Python pour utilisation sous LabVIEW

Théo Fernandez, LEGI, Juillet 2024
"""

def addition(A,B):
    return A+B

def soustraction(A,B):
    return A-B

def multiplication(A,B):
    return A*B

def division(A,B):
    return A/B
```

Code Python sous Spyder

Le format NetCDF (.nc)

Le format NetCDF (Network Common Data Form) est un format de fichier qui permet de stocker des données scientifiques multidimensionnelles sous la forme de tableaux.

Le format est binaire, "auto-documenté", signifiant qu'il existe un en-tête qui décrit la disposition des données dans le fichier, notamment des tableaux de données et des **variables**. Cet en-tête contient aussi une liste arbitraire de métadonnées se présentant sous la forme d'**attribut** de type nom/valeur.

Un fichier NetCDF contient donc notamment :

- **Dimensions** : nom, valeur (finie ou unlimited). Sera appliqué aux variables.
- **Variables** : nom, type, longueur, dimensions (nommées précédemment), attributs (ex: units, long_name, missing_value etc..).
- **Attributs globaux** : métadonnées pour l'ensemble du fichier, avec nom/valeur.

Écriture et lecture des fichiers NetCDF

Importer les bibliothèques NetCDF : **import netCDF4**

Pour l'écriture:

- **Dataset**: Créer un jeu de données NetCDF. Option : "w" : write, "r" : read, "r+" ou "a" : append

```
from netCDF4 import Dataset
ncfile = Dataset(namefile,mode='w',format='NETCDF4_CLASSIC')
```

- **.createDimension()** : Créer la dimension de l'ensemble de données à partir du nom et de la taille.

```
time_dim = ncfile.createDimension('time', None)   # unlimited axis
```

- **.createVariables()** : Créer une variable à partir d'un nom, d'un type et de la dimension.

```
time = ncfile.createVariable('time', np.float32, ('time',))
```

Pour la lecture:

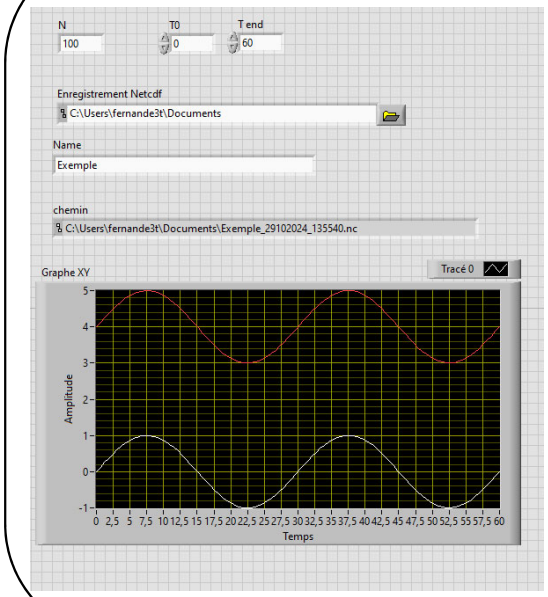
- **.variables.keys() :** Extrait les noms de variables avec leur dimension

```
variable_names = ncfile.variables.keys()
```

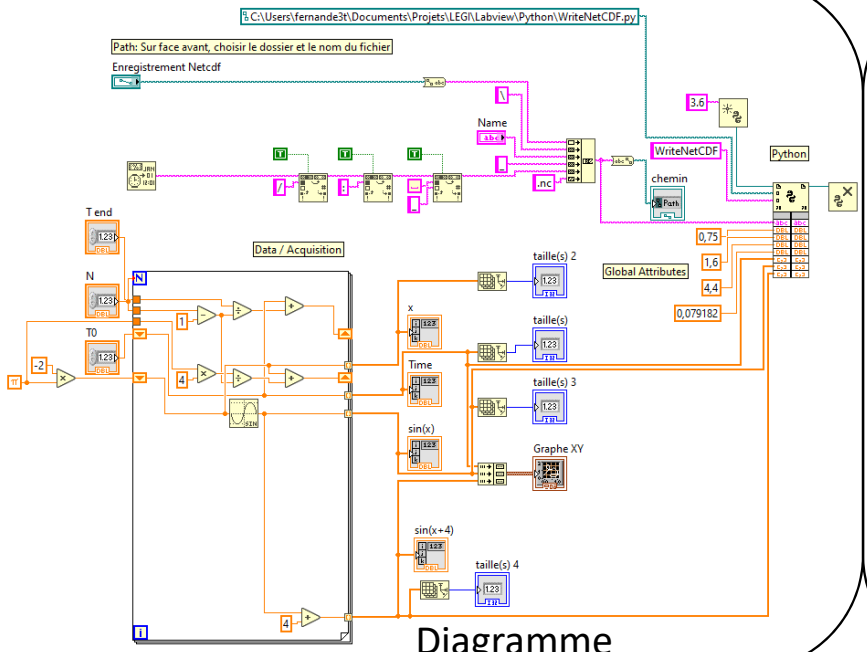
- **ncattrs() :** Extrait le nom des attributs

```
attribute_names = ncfile.ncattrs()
```

Exemple d'écriture des fichiers NetCDF



Face-avant



Diagramme

LabVIEW

```python
def WriteNetCDF(FileName,a,b,moyenne,ecarttype,time,Var1,Var2):

    import netCDF4 as nc4
    import numpy as np

    ncfile=nc4.Dataset(FileName,"w", format="NETCDF4_CLASSIC")
    ncfile.createDimension('time',None)

    Time = ncfile.createVariable('time',np.float32,('time',))
    Time.units = 's'
    Time.long_name = 'time'
    Time[:] = time

    var1 = ncfile.createVariable('Var1',np.float32,('time',))
    var1.units = 'V'
    var1[:] = Var1

    var2 = ncfile.createVariable('Var2',np.float32,('time',))
    var2.units = 'm/s'
    var2[:] = Var2

    ncfile.a = a
    ncfile.b = b
    ncfile.vitessemoyenne = moyenne
    ncfile.ecarttypevitesse = ecarttype
    ncfile.title = "Title"
    ncfile.subtitle = "Subtitle"

    # print(ncfile)
    ncfile.close()
```

Code Python sous Spyder

Exemple de lecture d'un fichier NetCDF


```python
from netCDF4 import Dataset

ncfile = Dataset('Exemple.nc',mode='r',format='NETCDF4_CLASSIC')


print(ncfile)

print(ncfile.variables.keys())
temps = ncfile.variables['time']
print(temps[:])
Var1 = ncfile.variables['Var1']
print(Var1[:])
Var2 = ncfile.variables['Var2']
print(Var2[:])
```

Programme Python

```
<class 'netCDF4._netCDF4.Dataset'>
root group (NETCDF4_CLASSIC data model, file format HDF5):
    a: 0.5
    b: 2.0
    moyenne: 4.0
    ecarttype: 0.08
    title: Title
    subtitle: Subtitle
    dimensions(sizes): time(400)
    variables(dimensions): float64 time(time), float64 Var1(time), float64 Var2(time)
    groups:
dict_keys(['time', 'Var1', 'Var2'])
[ 0.         0.15037594  0.30075188  0.45112782  0.60150376  0.7518797
  0.90225564  1.05263158  1.20300752  1.35338346  1.5037594   1.65413534
  1.80451128  1.95488722  2.10526316  2.2556391   2.40601504  2.55639098
  2.70676692  2.85714286  3.0075188   3.15789474  3.30827068  3.45864662
  3.60902256  3.7593985   3.90977444  4.06015038  4.21052632  4.36090226
  4.5112782   4.66165414  4.81203008  4.96240602  5.11278195  5.26315789
  5.41353383  5.56390977  5.71428571  5.86466165  6.01503759  6.16541353
  6.31578947  6.46616541  6.61654135  6.76691729  6.91729323  7.06766917
  7.21804511  7.36842105  7.51879699  7.66917293  7.81954887  7.96992481
  8.12030075  8.27067669  8.42105263  8.57142857  8.72180451  8.87218045
  9.02255639  9.17293233  9.32330827  9.47368421  9.62406015  9.77443609
  9.92481203 10.07518797 10.22556391 10.37593985 10.52631579 10.67669173
 10.82706767 10.97744361 11.12781955 11.27819549 11.42857143 11.57894737
 11.72932331 11.87969925 12.03007519 12.18045113 12.33082707 12.48120301
 12.63157895 12.78195489 12.93233083 13.08270677 13.23308271 13.38345865
 13.53383459 13.68421053 13.83458647 13.98496241 14.13533835 14.28571429
 14.43609023 14.58646617 14.73684211 14.88721805 15.03759398 15.18796992
 15.33834586 15.4887218  15.63909774 15.78947368 15.93984962 16.09022556
 16.2406015  16.39097744 16.54135338 16.69172932 16.84210526 16.9924812
 17.14285714 17.29323308 17.44360902 17.59398496 17.7443609  17.89473684
 18.04511278 18.19548872 18.34586466 18.4962406  18.64661654 18.79699248
 18.94736842 19.09774436 19.2481203  19.39849624 19.54887218 19.69924812
 19.84962406 20.         20.15037594 20.30075188 20.45112782 20.60150376
 20.7518797  20.90225564 21.05263158 21.20300752 21.35338346 21.5037594
 21.65413534 21.80451128 21.95488722 22.10526316 22.2556391  22.40601504
 22.55639098 22.70676692 22.85714286 23.0075188  23.15789474 23.30827068
 23.45864662 23.60902256 23.7593985  23.90977444 24.06015038 24.21052632
 24.36090226 24.5112782  24.66165414 24.81203008 24.96240602 25.11278195
 25.26315789 25.41353383 25.56390977 25.71428571 25.86466165 26.01503759
 26.16541353 26.31578947 26.46616541 26.61654135 26.76691729 26.91729323
 27.06766917 27.21804511 27.36842105 27.51879699 27.66917293 27.81954887
 27.96992481 28.12030075 28.27067669 28.42105263 28.57142857 28.72180451
 28.87218045 29.02255639 29.17293233 29.32330827 29.47368421 29.62406015
 29.77443609 29.92481203 30.07518797 30.22556391 30.37593985 30.52631579
 30.67669173 30.82706767 30.97744361 31.12781955 31.27819549 31.42857143
 31.57894737 31.72932331 31.87969925 32.03007519 32.18045113 32.33082707
```

Prompt sous Spyder

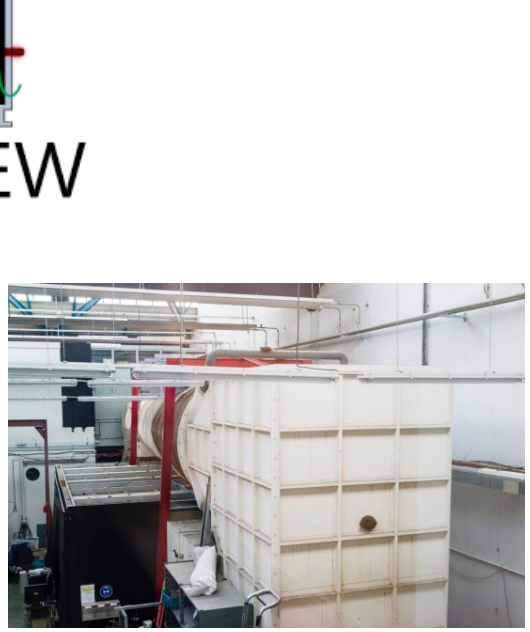
Sommaire

I. Contexte

II. Généralités
   1. Intégrer un code Matlab sur LabVIEW
   2. Appeler du code Python à partir de LabVIEW
   3. Écriture et lecture des fichiers NetCDF

III. Cas concret: la soufflerie
   1. Déplacement de la sonde
   2. Acquisition de données
   3. Enregistrement NetCDF Python

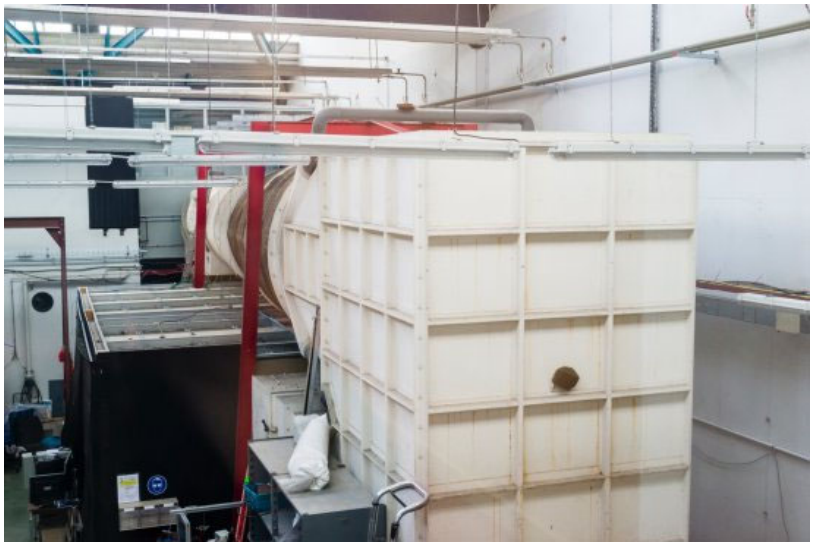
IV. Conclusion et perspective



Soufflerie: le grand instrument

La soufflerie à bas niveau de turbulence du LEGI :

- Etude des écoulements turbulents (mono et diphasiques)
- Circuit fermé
- Dimensions : 5 mètres de haut, 16 mètres de long, et 2.5 mètres de large.
- Veine d'essais mesure 4 mètres de long (section carrée (0.75x0.75 m^2)).
- Vitesses peuvent atteindre 45 m/s dans la veine d'essai.
- Intensité turbulente résiduelle < 0.1%.



Grille active générant de la turbulence

Injection de gouttelettes

Calibration et mesure :

Anémomètres soniques, sonde de pitot, anémomètre à fil chaud

Code Arduino

Déplacement vertical de la sonde à fil chaud :

Carte Arduino Uno

Réglage des vitesses et du déplacement maximal :

- ✓ long max_steps = 100000;
- ✓ long homing_fast = 5000;
- ✓ long homing_slow = 1;

Utilisation de la librairie AccelStepper

Ecriture de paramètres (positions de la sonde) sur la liaison série :

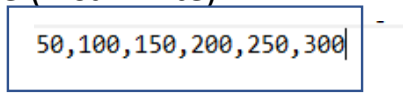
- ✓ **Homing :** *Serial.println("Homing");*
- ✓ **En déplacement :** *Serial.println("Moving");*
- ✓ **Déplacement terminé :** *Serial.println("Completed");*

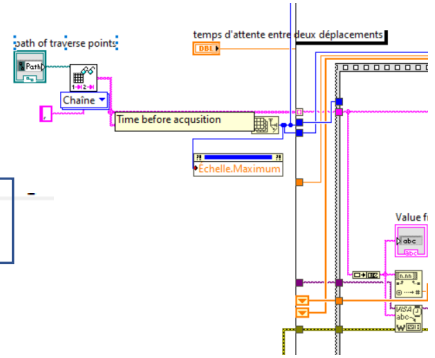
Code LabView

Déplacement vertical de la sonde à fil chaud :

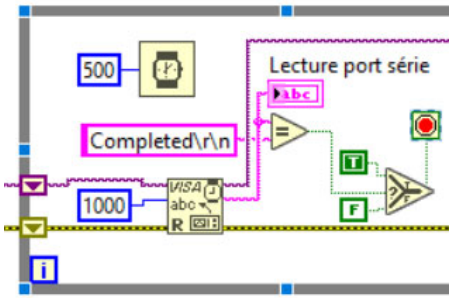
LabView :

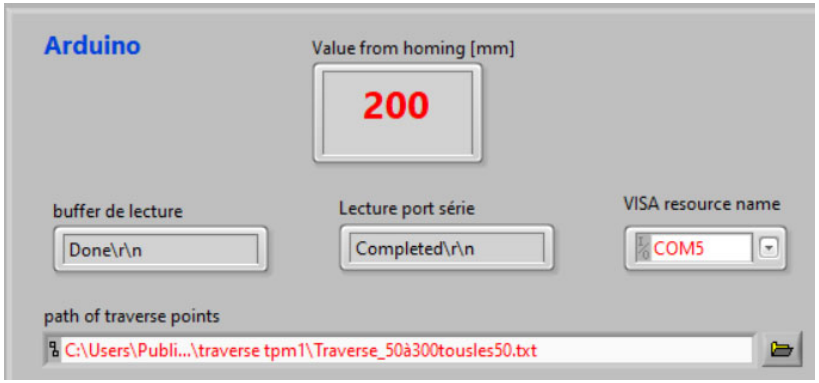
- Envoi de la série de déplacements (fichier) par écriture sur liaison série (Visa Write)
- Ajout d'un temps d'attente entre chaque déplacement





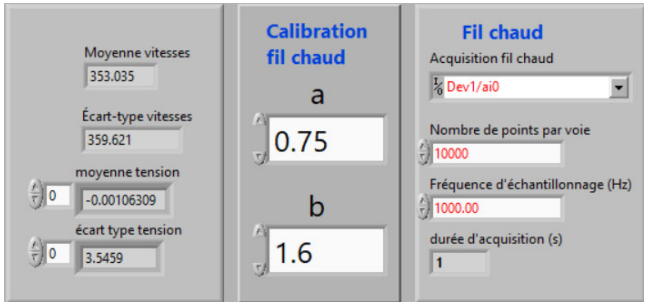
- Récupération des instructions série de l'Arduino (Visa Read)
- Affichage de la position de la sonde

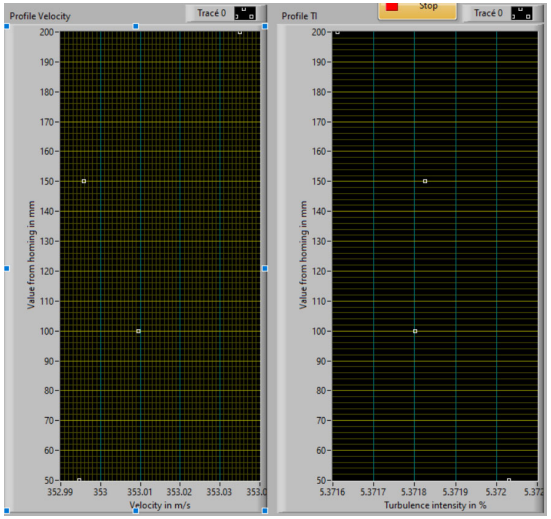




Code LabView

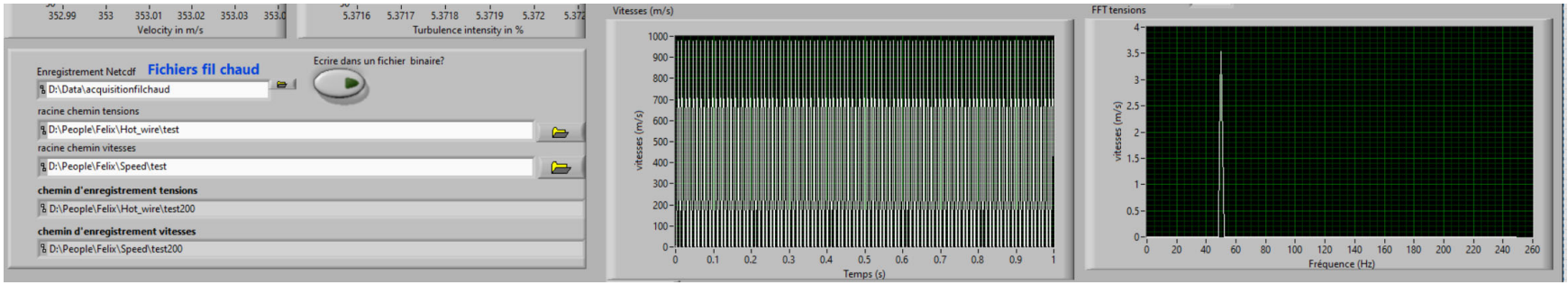
LabView :





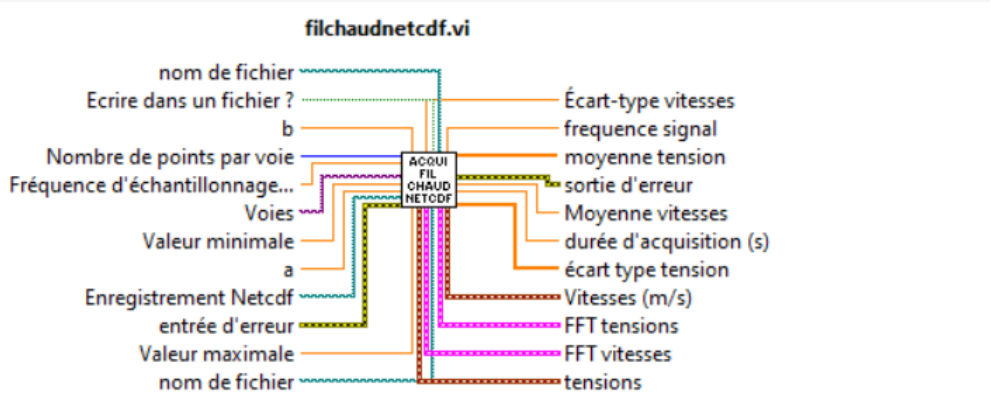
A chaque arrêt :

- Acquisition du signal du fil chaud
- Affichage tensions, vitesses et FFT
- Graphes vitesses et intensité turbulente en fonction de la position
- Enregistrement binaire et Netcdf



LabVIEW: sous VI

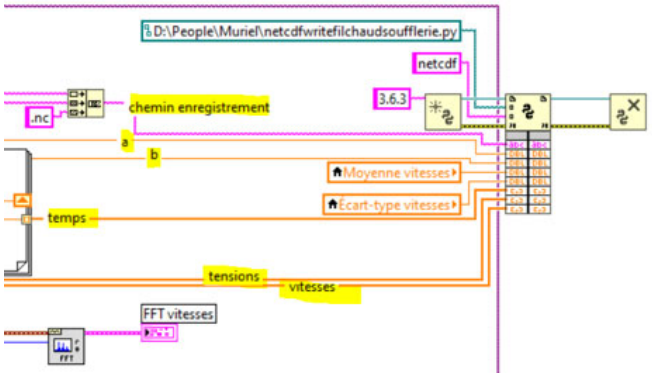
Le sous VI d'acquisition du fil chaud et d'enregistrement en format NetCDF



M.Lagauzère juin 2024
Ce VI est utilisé sur la soufflerie du Legi pour une acquisition d'une voie de fil chaud (voie ai0 de la carte NI USB-6356). Il réalise une acquisition finie de tension et calcule la vitesse en fonction des paramètres a et b entrés par l'utilisateur.
Il calcule également la vitesse moyenne ainsi l'écart type de la vitesse.
Il enregistre le fichier de mesure de tensions et celui de vitesses sous une forme binaire.

Il enregistre également les valeurs de tensions, de vitesses ainsi que les paramètres a et b, la vitesse moyenne et l'écart type de vitesses sous un fichier netcdf (nc)

LabVIEW: diagramme




```
Python 3.6.7 Shell
File  Edit  Shell  Debug  Options  Window  Help
Python 3.6.7 (v3.6.7:6ec5cf24b7, Oct 20 2018, 13:35:33) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
 RESTART: C:\Users\Public\déplacement sonde verticale et acquisition fils chauds\netcdfreadfilchaudsoufflerie.py
<class 'netCDF4._netCDF4.Dataset'>
root group (NETCDF4_CLASSIC data model, file format HDF5):
    a: 0.75
    b: 1.6
    vitessemoyenne: 4.520052583816302
    ecarttypevitesse: 0.018929016156103474
    positionenmm: 200.0
    title: acquisition fil chaud
    subtitle: soufflerie
    dimensions(sizes): time(500)
    variables(dimensions): float32 time(time), float32 tensionsfilchaud(time), float32 vitessesfilchaud(time)
    groups:
odict_keys(['time', 'tensionsfilchaud', 'vitessesfilchaud'])
[0.    0.002 0.004 0.006 0.008 0.01  0.012 0.014 0.016 0.018 0.02  0.022
 0.024 0.026 0.028 0.03  0.032 0.034 0.036 0.038 0.04  0.042 0.044 0.046
 0.048 0.05  0.052 0.054 0.056 0.058 0.06  0.062 0.064 0.066 0.068 0.07
 0.072 0.074 0.076 0.078 0.08  0.082 0.084 0.086 0.088 0.09  0.092 0.094
 0.096 0.098 0.1   0.102 0.104 0.106 0.108 0.11  0.112 0.114 0.116 0.118
 0.12  0.122 0.124 0.126 0.128 0.13  0.132 0.134 0.136 0.138 0.14  0.142
 0.144 0.146 0.148 0.15  0.152 0.154 0.156 0.158 0.16  0.162 0.164 0.166
 0.168 0.17  0.172 0.174 0.176 0.178 0.18  0.182 0.184 0.186 0.188 0.19
 0.192 0.194 0.196 0.198 0.2   0.202 0.204 0.206 0.208 0.21  0.212 0.214
 0.216 0.218 0.22  0.222 0.224 0.226 0.228 0.23  0.232 0.234 0.236 0.238
 0.24  0.242 0.244 0.246 0.248 0.25  0.252 0.254 0.256 0.258 0.26  0.262
 0.264 0.266 0.268 0.27  0.272 0.274 0.276 0.278 0.28  0.282 0.284 0.286
 0.288 0.29  0.292 0.294 0.296 0.298 0.3   0.302 0.304 0.306 0.308 0.31
 0.312 0.314 0.316 0.318 0.32  0.322 0.324 0.326 0.328 0.33  0.332 0.334
 0.336 0.338 0.34  0.342 0.344 0.346 0.348 0.35  0.352 0.354 0.356 0.358
 0.36  0.362 0.364 0.366 0.368 0.37  0.372 0.374 0.376 0.378 0.38  0.382
 0.384 0.386 0.388 0.39  0.392 0.394 0.396 0.398 0.4   0.402 0.404 0.406
 0.408 0.41  0.412 0.414 0.416 0.418 0.42  0.422 0.424 0.426 0.428 0.43
 0.432 0.434 0.436 0.438 0.44  0.442 0.444 0.446 0.448 0.45  0.452 0.454
 0.456 0.458 0.46  0.462 0.464 0.466 0.468 0.47  0.472 0.474 0.476 0.478
 0.48  0.482 0.484 0.486 0.488 0.49  0.492 0.494 0.496 0.498 0.5   0.502
 0.504 0.506 0.508 0.51  0.512 0.514 0.516 0.518 0.52  0.522 0.524 0.526
 0.528 0.53  0.532 0.534 0.536 0.538 0.54  0.542 0.544 0.546 0.548 0.55
 0.552 0.554 0.556 0.558 0.56  0.562 0.564 0.566 0.568 0.57  0.572 0.574
 0.576 0.578 0.58  0.582 0.584 0.586 0.588 0.59  0.592 0.594 0.596 0.598
 0.6   0.602 0.604 0.606 0.608 0.61  0.612 0.614 0.616 0.618 0.62  0.622
 0.624 0.626 0.628 0.63  0.632 0.634 0.636 0.638 0.64  0.642 0.644 0.646
 0.648 0.65  0.652 0.654 0.656 0.658 0.66  0.662 0.664 0.666 0.668 0.67
 0.672 0.674 0.676 0.678 0.68  0.682 0.684 0.686 0.688 0.69  0.692 0.694
 0.696 0.698 0.7   0.702 0.704 0.706 0.708 0.71  0.712 0.714 0.716 0.718
 0.72  0.722 0.724 0.726 0.728 0.73  0.732 0.734 0.736 0.738 0.74  0.742
 0.744 0.746 0.748 0.75  0.752 0.754 0.756 0.758 0.76  0.762 0.764 0.766
 0.768 0.77  0.772 0.774 0.776 0.778 0.78  0.782 0.784 0.786 0.788 0.79
 0.792 0.794 0.796 0.798 0.8   0.802 0.804 0.806 0.808 0.81  0.812 0.814
 0.816 0.818 0.82  0.822 0.824 0.826 0.828 0.83  0.832 0.834 0.836 0.838
 0.84  0.842 0.844 0.846 0.848 0.85  0.852 0.854 0.856 0.858 0.86  0.862
 0.864 0.866 0.868 0.87  0.872 0.874 0.876 0.878 0.88  0.882 0.884 0.886
 0.888 0.89  0.892 0.894 0.896 0.898 0.9   0.902 0.904 0.906 0.908 0.91
 0.912 0.914 0.916 0.918 0.92  0.922 0.924 0.926 0.928 0.93  0.932 0.934
 0.936 0.938 0.94  0.942 0.944 0.946 0.948 0.95  0.952 0.954 0.956 0.958
 0.96  0.962 0.964 0.966 0.968 0.97  0.972 0.974 0.976 0.978 0.98  0.982
 0.984 0.986 0.988 0.99  0.992 0.994 0.996 0.9981
```

- Formatage du nom du fichier Netcdf sous la forme ‘acquisitionfilchaud+date/heure+position (mm)’

- Création du fichier d’écriture :

✓ des métadonnées (a, b, vitesse moyenne, écart type vitesse, position)

✓ des colonnes de temps, tensions et vitesses

Code LabVIEW

Déplacement vertical de la sonde à fil chaud :

Etapes:

1 Retour à la position origine (homing) : attente du retour ‘Homing’ sur la ligne série

2 Envoi du déplacement : attente du retour ‘Completed’ sur la ligne série

3 Temps d’attente pour stabilisation de l’écoulement

4 Acquisition de la mesure de tension (fil chaud)

5 Calcul vitesse (vitesse moyenne et écart type), FFT vitesses

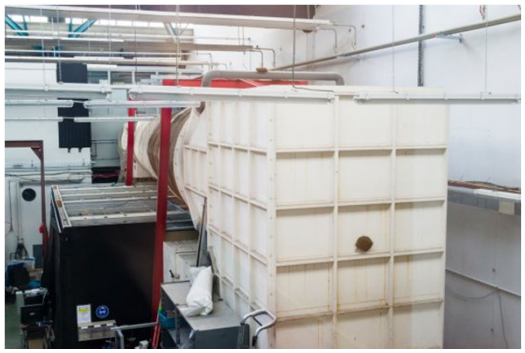
6 Affichage graphes vitesse et intensité turbulente en fonction de la position à partir de la position d’origine

7 Enregistrement des données (binaire et NetCDF)

Utilisation d’une boucle for (nombre de déplacements lus dans le fichier de commande)

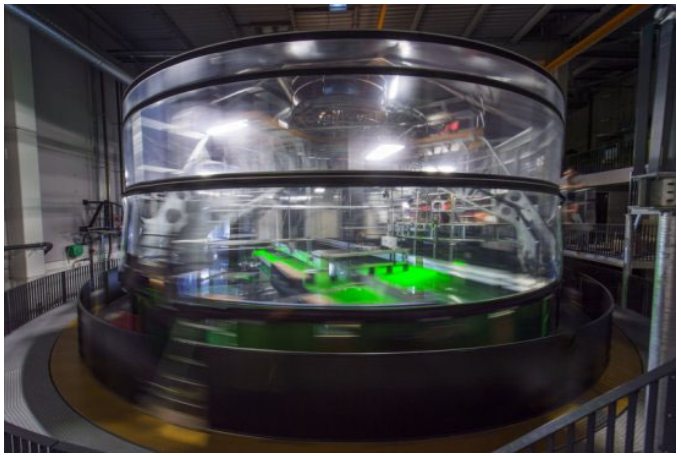
et d’une structure séquence

Sommaire



Conclusion et perspective

- Intégration simple de Python pour l'écriture de fichier en NetCDF
- Utilisation de ce type d'enregistrement sur la plateforme Coriolis (18 voies d'acquisition en acquisition continue)
- Fichiers mieux documentés et plus uniformisés



- Des fonctions similaires pour matlab
- Des fonctions similaires pour d'autres formats comme le HDF5 (.h5)
- Intégration sur d'autres expériences

Appel de fonctions Python / Matlab à partir de LabVIEW

Un exemple avec la création de fichier NetCDF

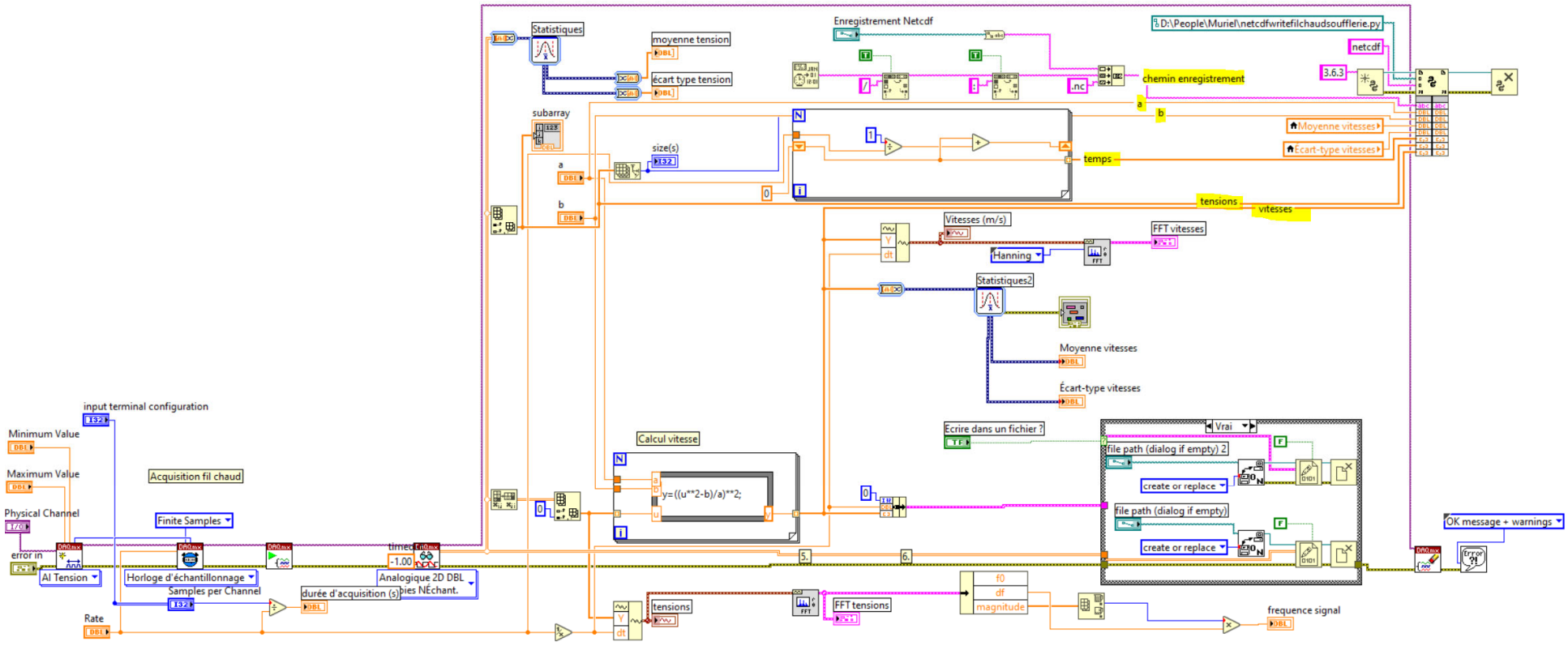
Muriel Lagauzère, Théo Fernandez

Laboratoire des Écoulements Géophysiques et Industriels (LEGI), UMR 5519, Grenoble, France

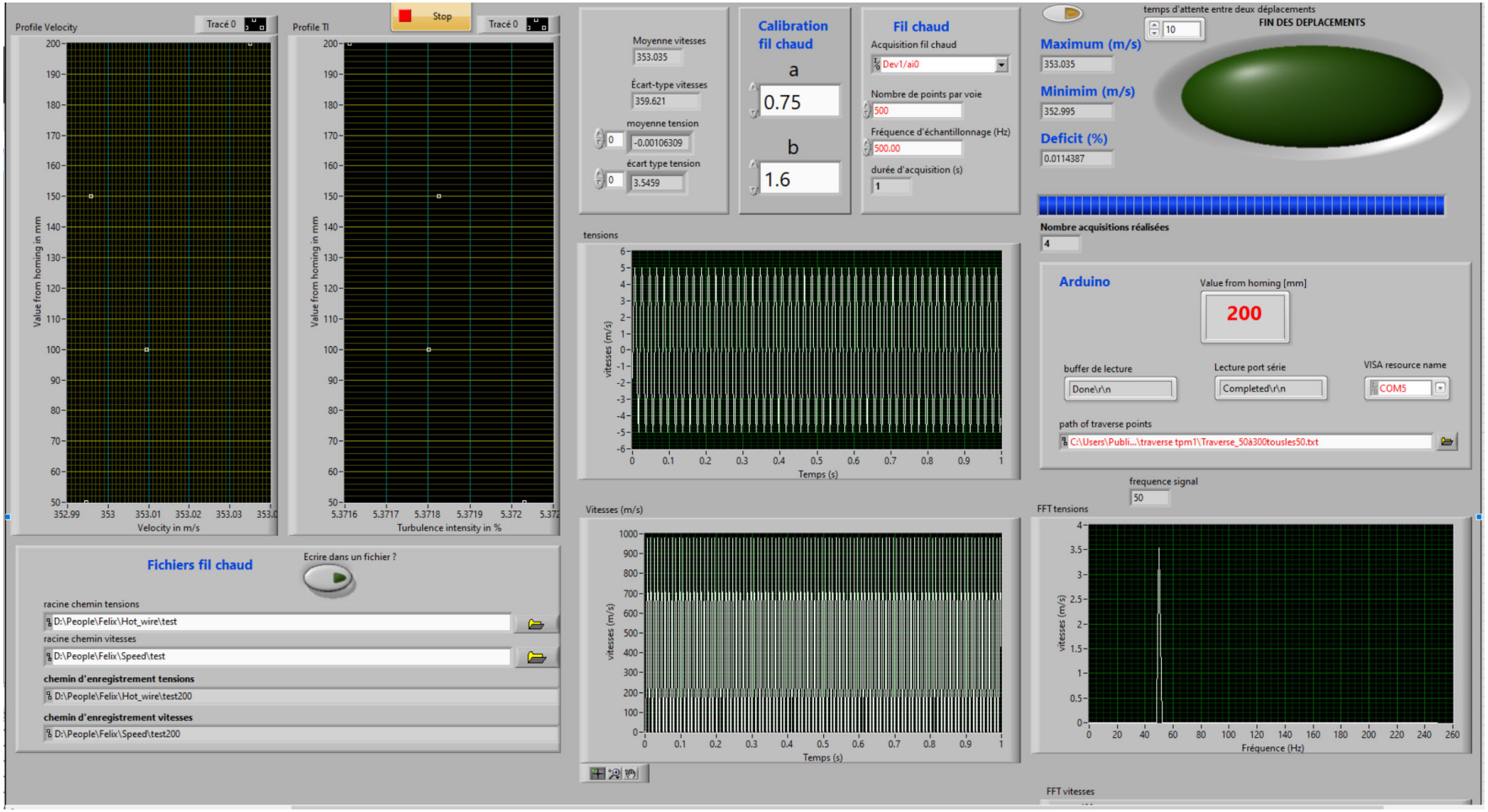
Rencontre AlpesVIEW/CNRS, Jeudi 14 Novembre, 2024, Institut Néel, Grenoble, France

Annexes

LabVIEW: diagramme



LabVIEW: Face avant



Écriture et lecture des fichiers NetCDF

Avec Matlab:

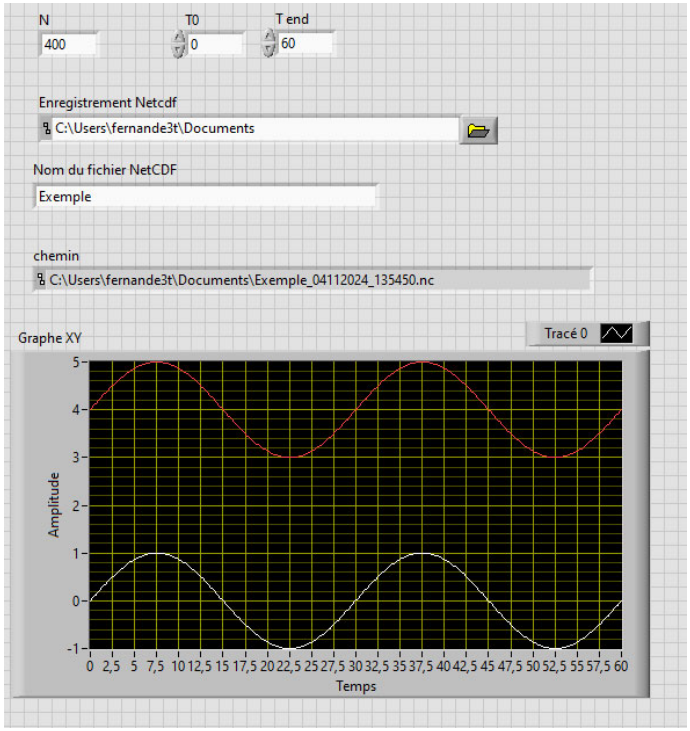
Pour l'écriture:

- **nccreate** : Créer une variable dans un fichier NetCDF
- **ncwrite** : Écrire des données dans un fichier netCDF
- **ncwriteatt** : Écrire un attribut dans un fichier netCDF

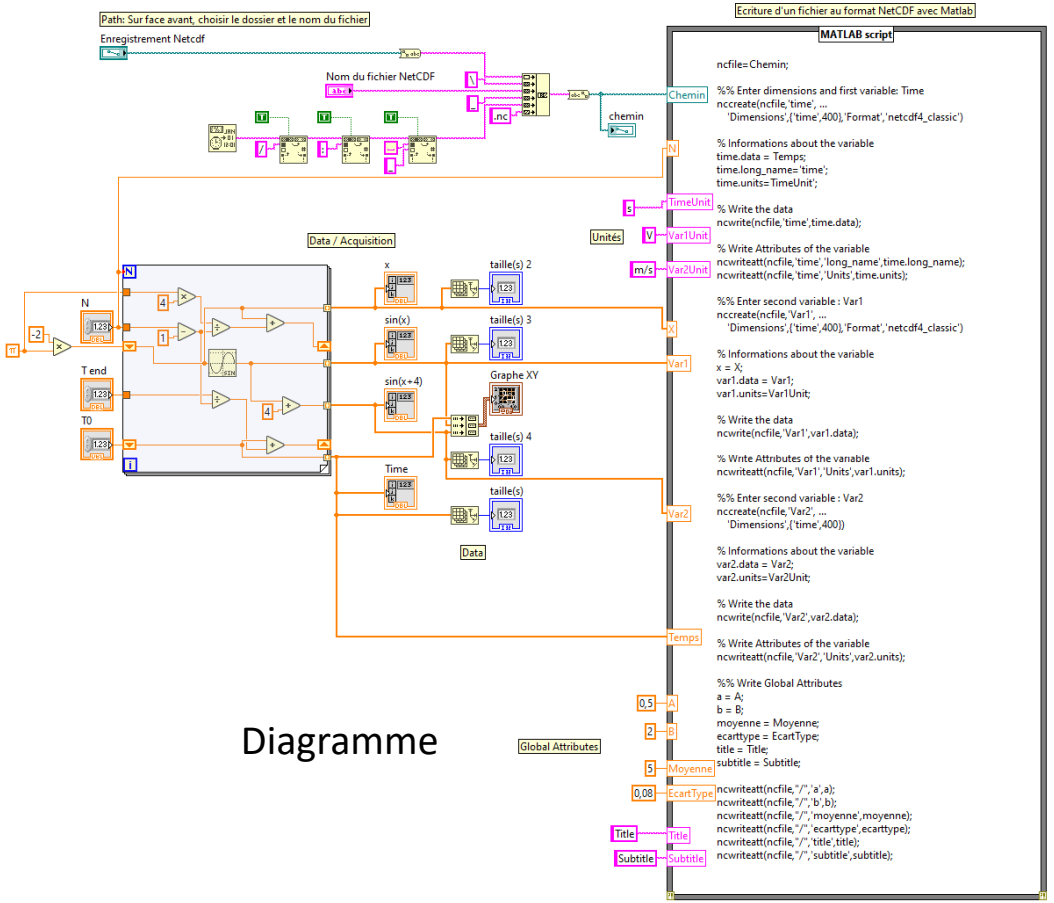
Pour la lecture:

- **ncdisp** : Afficher le contenu de la source de donnée netCDF dans la Command Window
- **ncinfo** : Renvoyer des informations sur la source de données netCDF
- **ncread** : Lire les données d'une variable dans une source de données netCDF
- **ncreadatt** : Lire l'attribut de la source de données netCDF

Exemple d'écriture des fichiers NetCDF



Face-avant



Diagramme

Exemple de lecture d'un fichier NetCDF


```
clear; close all; clc

% Let the user choose
[FullFileName, PathName1] = uigetfile({'*.nc'},'Select the measurement files in NetCDF (*.nc)', 'MultiSelect', 'off'); % Selection of the files
ncfile = [PathName1,FullFileName];

% Get the data in a structure : Filchaud
Data = ncinfo(ncfile);

% Display everything that is in the file
ncdisp(ncfile);

% To get the names of the variables automatically in a cell
for i = 1:length(Data.Attributes) ...

% Let the user choose which variable to choose
[Idx_Attrb,tf] = listdlg('ListString',NamesAllAttrb,'PromptString',{'Select the Attributes you want to extract',''}); clear tf; % List for selection of the profiles
for i=1:length(Idx_Attrb) ...

% Get the time anyway
Time = ncread(ncfile,'time');

% To get the names of the variables automatically in a cell
for i = 2:length(Data.Variables) ...

% Let the user choose which variable to choose
[Idx_Var,tf] = listdlg('ListString',NamesAllVar,'PromptString',{'Select the variables you want to analyze',''}); clear tf; % List for selection of the profiles
for i=1:length(Idx_Var) ...

%%  Plots : All in one

figure();

for i=1:length(Variables) ...

xlabel(['Time [s]']); ylabel(['Value']);
box on; grid on; legend off; legend show;


clear FullFileName PathName1 Idx_Attrb Idx_Var i NamesAllAttrb NamesAllVar
disp('Program run successfully');
```

Programme Matlab

```
>> ncdisp(ncfile)
Source:
           C:\Users\fernande3t\Documents\Projets\LEGI\Matlab\NetCDF\Exemple.nc
Format:
           netcdf4_classic
Global Attributes:
           a          = 0.5
           b          = 2
           moyenne    = 4
           ecarttype  = 0.08
           title      = 'Title'
           subtitle   = 'Subtitle'
Dimensions:
           time = 400
Variables:
    time
           Size:       400x1
           Dimensions: time
           Datatype:   double
           Attributes:
                       long_name = 'time'
                       Units     = 's'
    Var1
           Size:       400x1
           Dimensions: time
           Datatype:   double
           Attributes:
                       Units = 'V'
    Var2
           Size:       400x1
           Dimensions: time
           Datatype:   double
           Attributes:
                       Units = 'm/s'
```

Prompt sous Matlab

Écriture et lecture des fichiers HDF5

Le **Hierarchical Data Format** (HDF) est un ensemble de formats de fichiers permettant de sauvegarder et de structurer des fichiers contenant de très grandes quantités de données. Un fichier HDF est un conteneur de fichiers. Il existe principalement deux formats HDF : HDF4 et HDF5. Les fichiers HDF ont de nombreuses extensions : .hdf, .h4, .hdf4, .he4 (pour HDF4) .h5, .hdf5 et .he5 (pour HDF5). Le HDF5 améliore le HDF4. Il simplifie la structure des fichiers pour proposer seulement deux types d'objets :

- les ensembles de données (**datasets**), qui sont des tableaux multidimensionnels contenant des données d'un même type (prédéfini ou dérivé) ;
- les groupes (**groups**), qui contiennent, ou regroupent, des ensembles de données et d'autres groupes.

L'utilisateur peut aussi définir une structure d'attribut pour ajouter des informations (des meta-données) sur chaque objet. Cette structuration en ensembles de données et groupes donne aux fichiers HDF5 une structure hiérarchique (en l'occurrence arborescente), un peu comme un système de fichiers. Cette analogie avec un système de fichiers est aussi présente dans les conventions de nommage des objets du fichiers : le nom complet d'un objet du fichier HDF5 est construit comme un chemin dans un système de fichiers. NetCDF utilise par exemple HDF5 depuis la version 4.

Écriture et lecture des fichiers HDF5

Avec Matlab:

Pour l'écriture:

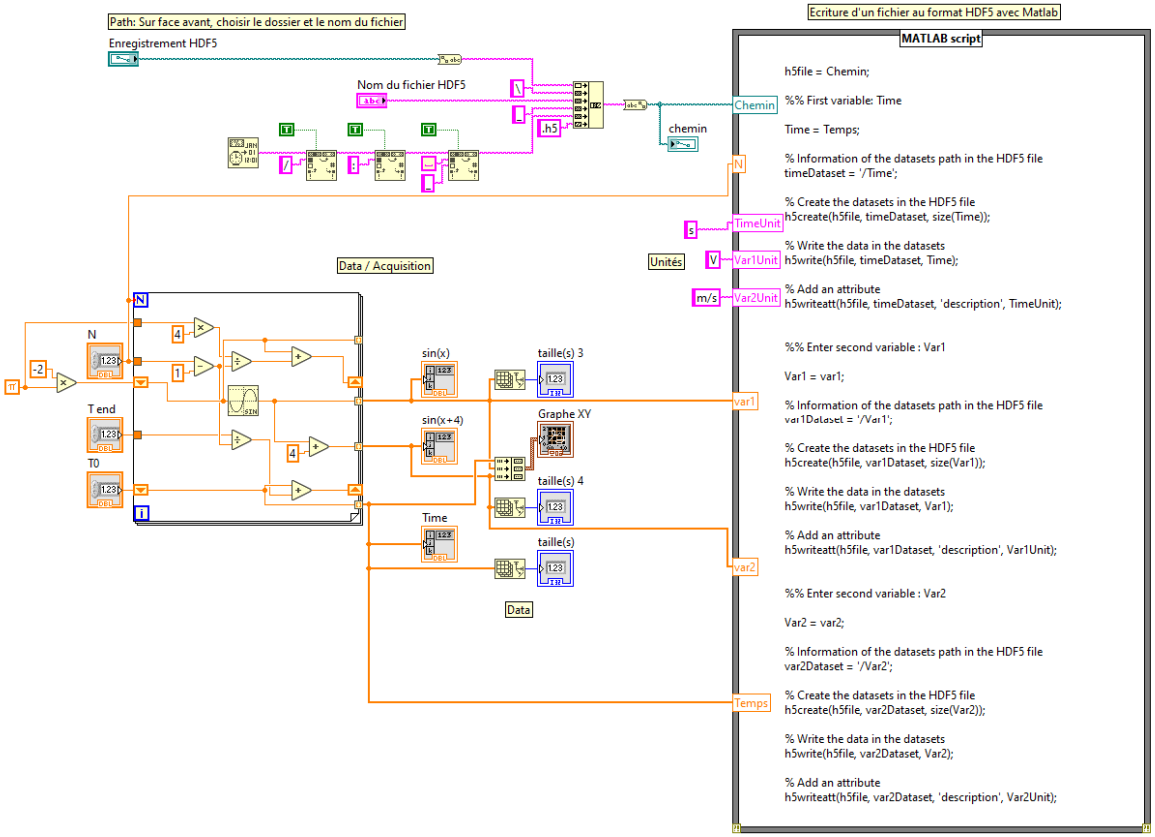
- **h5create** : Créer un dataset HDF5 h5info : Informations sur le fichier HDF5
- **h5write** : Écriture de données dans un dataset HDF5
- **h5writeatt** : Écriture d'un attribut dans un fichier HDF5

Pour la lecture:

- **h5disp** : Afficher le contenu d'un fichier HDF5
- **h5info** : Informations sur le fichier HDF5
- **h5read** : Lire les datasets de données HDF5
- **h5readatt** : Lecture d'un attribut dans un fichier HDF5

Écriture et lecture des fichiers HDF5

Avec Matlab:




```
>> h5disp(h5file)
HDF5 Exemple.h5
Group '/'
    Dataset 'Time'
        Size:  1x100
        MaxSize:  1x100
        Datatype:   H5T_IEEE_F64LE (double)
        ChunkSize:  []
        Filters:  none
        FillValue:  0.000000
        Attributes:
            'description':  'Time in [s]'
    Dataset 'Var1'
        Size:  1x100
        MaxSize:  1x100
        Datatype:   H5T_IEEE_F64LE (double)
        ChunkSize:  []
        Filters:  none
        FillValue:  0.000000
        Attributes:
            'description':  'Sinus with Amplitude [-1, 1]'
    Dataset 'Var2'
        Size:  1x100
        MaxSize:  1x100
        Datatype:   H5T_IEEE_F64LE (double)
        ChunkSize:  []
        Filters:  none
        FillValue:  0.000000
        Attributes:
            'description':  'Sinus with Amplitude [-2, 2]'
```

Écriture et lecture des fichiers HDF5

Avec Python:

Importer les bibliothèques HDF5 : **import h5py**

Pour l'écriture:

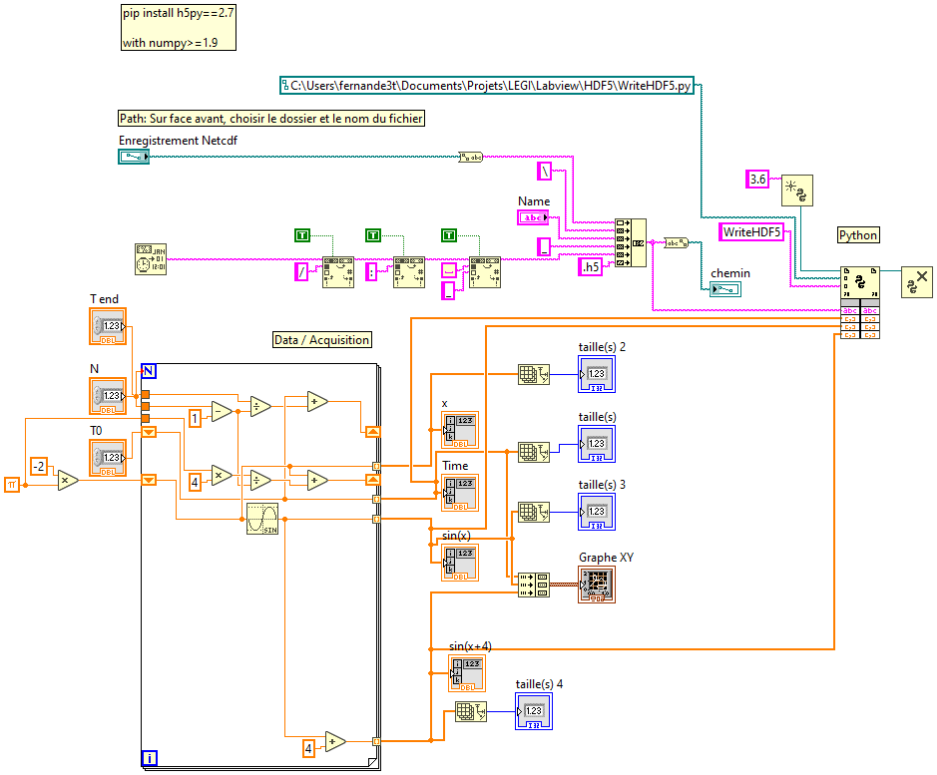
- **h5py.File** : Créer un dataset HDF5 vide. Option : "w" : write, "r" : read, "a" : append
- **.create_dataset()** : Crée un dataset à partir d'un nom, d'un type et de dimensions.
- **.attrs** : Définir ou obtenir des attributs pour un dataset ou le fichier lui-même

Pour la lecture:

- **.variables.keys() :** Extrait les noms de variables avec leur dimension

Écriture et lecture des fichiers HDF5

Avec Python:




```python
def WriteHDF5(FileName, time, Var1, Var2):
    import h5py

    h5file = h5py.File(FileName, "w")

    Time_dataset = h5file.create_dataset('time', data=time, dtype='f4')
    Time_dataset.attrs['units'] = 's'

    Var1_dataset = h5file.create_dataset('Var1', data=Var1, dtype='f4')
    Var1_dataset.attrs['units'] = 'V'

    Var2_dataset = h5file.create_dataset('Var2', data=Var2, dtype='f4')
    Var2_dataset.attrs['units'] = 'm/s'

    h5file.close()
```