

# PFE. Reconstruction de champ de vent.

Henri LEROY

## Résumé

Dans le contexte de l'éolien, la modélisation des champs de vent est un enjeu important pour savoir notamment où placer les éoliennes afin d'optimiser leur productivité. Une problématique commune est alors la reconstruction du champ de vent sur une zone étant donné des mesures ponctuelles de ce dernier. Plusieurs approches sont possibles pour aborder ce problème. Une idée simple consiste à exploiter des méthodes d'interpolation multivariée. Une autre approche revient à résoudre un problème de minimisation sous contrainte, permettant de prendre en compte les propriétés physiques. Le but de ce PFE est de s'initier aux méthodes numériques pour cette problématique.

## Première partie

# Récupération & visualisation des données.

Dans un premier temps, l'objectif sera de récupérer des données sur les vents afin d'avoir une base de travail. Nous essayerons également de visualiser ses données afin de se familiariser avec le sujet.

## 1 Choix de la base de données.

### 1.1 Variables à considérer

Plusieurs options sont disponibles pour accéder à des données sur les vents. Avant de commencer notre travail, nous devons donc choisir un moyen d'obtenir les données.

Tout d'abord, il est intéressant de réfléchir aux variables à avoir sur les vents pour notre travail. Dans notre cas, 2 valeurs seront nécessaires :

1. La **direction** du vent.
2. La **vitesse** du vent.

Avec ces 2 valeurs, nous pourrions représenter le champ de vecteurs et commencer notre travail d'interpolation. On doit donc trouver une base de données permettant d'accéder à ces champs. De plus, on doit penser au pas de temps et d'espace utilisé. En effet, l'espace sera considéré comme une grille en 2D avec des points espacés d'une certaine longitude et latitude. Un maillage assez fin permettra d'obtenir une simulation plus réaliste. Enfin, les données obtenues devront être espacées d'une certaine durée. Afin d'obtenir l'évolution des vents sur une journée, un mois ou une année, nous devrions avoir l'évolution de ces mesures du vent dans le temps. Bien que cette option ne soit pas indispensable, elle pourrait être intéressante à observer dans le but de trouver le point optimal pour placer une éolienne. Il faudrait considérer l'évolution du vent au cours du temps pour optimiser la position de l'éolienne.

### 1.2 Choix d'une base de données

Avant d'interpoler nos données, nous devons trouver une source permettant d'obtenir des données réelles ayant les caractéristiques présentées précédemment.

- Dans un premier temps, j'ai essayé d'utiliser la base de données WRDB (Wind Resource Database) <https://wrdb.nrel.gov>. Sur cette base de données, on peut obtenir la vitesse et la direction du vent à plusieurs altitudes (de 10m à 160m de hauteur). Ces informations pourront être utiles pour effectuer une interpolation en 3D. On pourra commencer par fixer une altitude avant d'améliorer notre projet en ajoutant une dimension. Pour la localisation géographique, plusieurs pays sont disponibles. Pour commencer, j'ai choisi de travailler sur la zone

Ukraine. Cette zone m'a semblé assez pertinente dans la mesure où les données sont disponibles de 2000 à 2022 soit sur une large période de temps. De plus, la résolution spatiale (pas dans l'espace) est de 2km et la résolution temporelle (pas de temps) est de 5min ce qui me paraît assez bas comme valeur. La simulation obtenue pourra donc être réaliste. De plus, la zone étudiée (en bleu sur la figure ci-après) possède un profil topologique varié avec des chaînes montagneuses et la mer Noire ce qui pourrait permettre, d'après moi, une étude plus intéressante des vents.

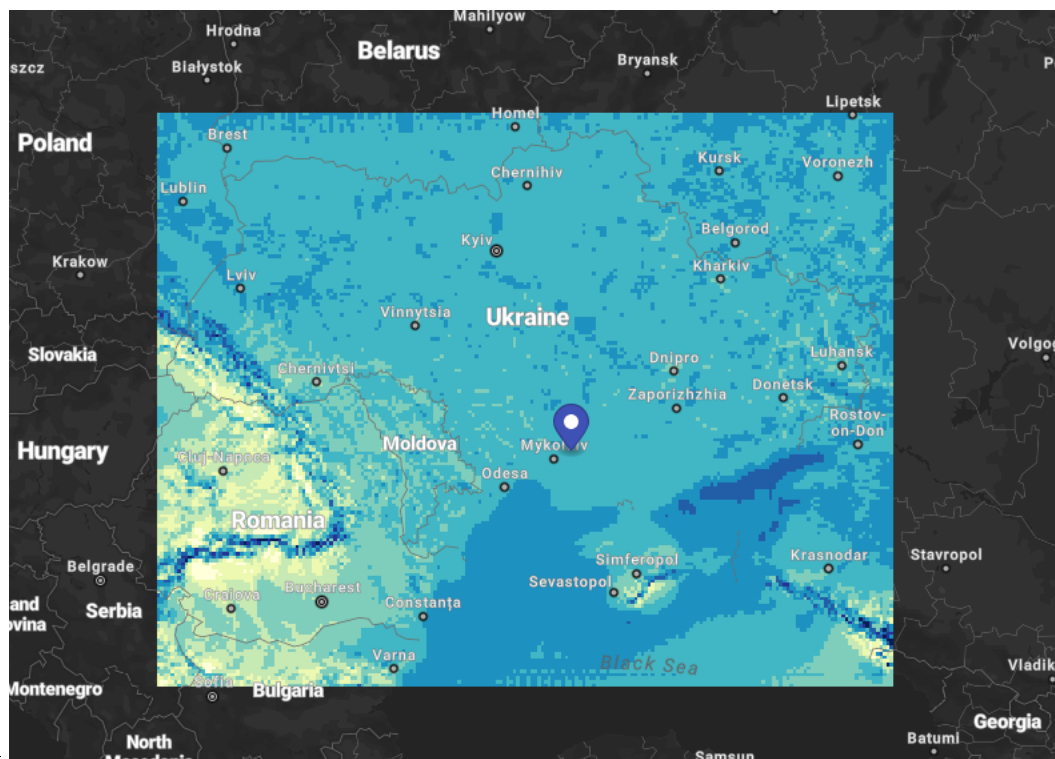


FIGURE 1 — Définition de la zone géographique étudiée (en bleu).

Cependant, le principal inconvénient de cette base de données réside dans la manière de récupérer les données. En effet, on doit indiquer une latitude et une longitude avant d'obtenir les données en rapport avec ce point. De plus, il faut indiquer une adresse mail ce qui pourrait rendre plus compliqué l'obtention automatique des données. La création du jeu de données entier serait donc assez contraignant ce qui m'a poussé à m'intéresser à d'autres bases de données disponibles. J'ai donc décidé de choisir une autre base de données plus adaptée.

- Par la suite, j'ai donc essayé d'utiliser la base de données de Global Wind Atlas <https://globalwindatlas.info/en/>. Après avoir réussi à recevoir le fichier de données, j'ai dû le convertir pour passer d'un fichier .tif à un jeu de données compréhensible. Malheureusement, après avoir réussi à convertir ce fichier, je me suis rendu compte que la base de données ne contenait aucune information sur la direction du vent. J'ai donc pu observer la vitesse du vent partout en France pour plusieurs altitudes, sans avoir d'indication sur sa direction. Par conséquent, j'ai dû trouver une autre base de données à exploiter.

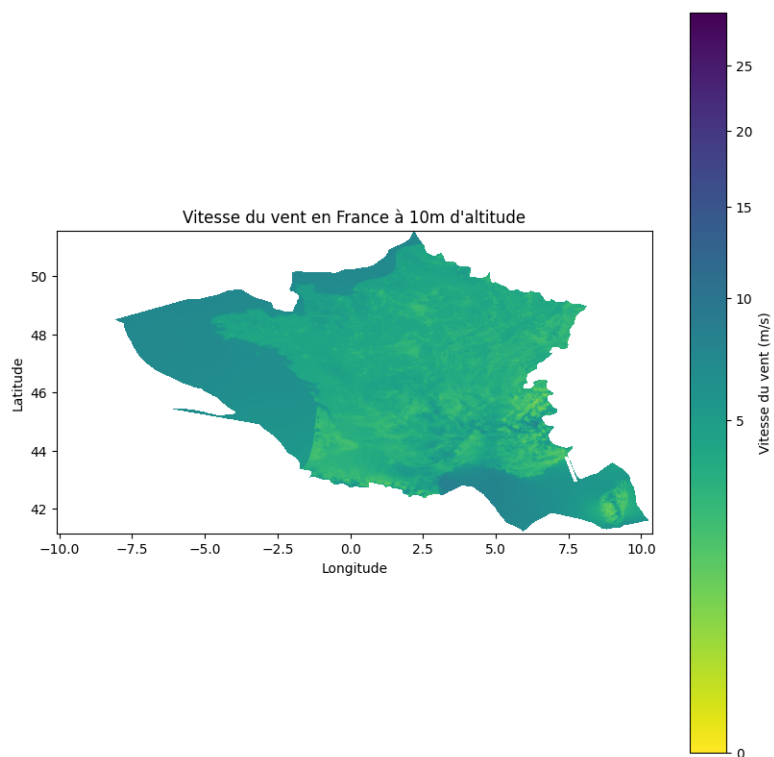


FIGURE 2 –

Projection des vitesses de vents en France via Global Wind Atlas. Seule la vitesse des vents est indiquée, on ne peut pas avoir sa direction ce qui limite son utilité dans notre projet.

- Après avoir tenté d'utiliser 2 bases de données différentes, en vain, j'ai décidé d'utiliser la base de données stormglass.io <https://stormglass.io/wind-api/> possédant plusieurs informations climatiques utiles. Cette API est bien documentée et complète ce qui devrait nous permettre de l'utiliser. L'inconvénient majeur de cette api est le nombre de requêtes disponibles par jour. Dans sa version gratuite, seules 10 requêtes sont autorisées par jour. De plus, cette base de données n'accepte que les requêtes à une latitude et une longitude donnée. Il faudrait donc effectuer plusieurs requêtes pour avoir une vision globale d'un terrain. Cette contrainte nous empêche d'avoir facilement des données. J'ai donc dû poursuivre ma recherche d'une base de données idéale.
- J'ai finalement trouvé une base de données exploitable pour notre projet sur le site <https://open-meteo.com/en/docs/historical-weather-api> en exploitant l'API « Historical Weather ». Cette API me permet d'obtenir la direction et la vitesse du vent (en km/h) à 10m et à 100m d'altitude mais aussi bien d'autres variables (température, humidité...). De plus, on peut obtenir ces informations pour une liste de coordonnées spécifiées ce qui nous permet de quadriller une zone en connaissant sa latitude et sa longitude. On a également une variable temporelle : on obtient les informations demandées pour chaque heure de la journée sur une plage temporelle donnée. On peut également passer une liste de coordonnées en entrée pour obtenir les variables étudiées sur plusieurs points. J'ai donc choisi cette base de données pour ce projet. Toutefois, le nombre de requêtes par jour est limité à 600. Je ne pense pas que cette valeur soit limitante mais il faudra éviter d'effectuer des requêtes en boucle pour éviter tout désagrément.

## 2 Première étude du jeu de données.

Après avoir choisi le jeu de données qui sera étudié dans la première partie de ce projet, nous allons effectuer une première étude afin de se familiariser avec lui.

Pour commencer, j'ai essayé d'obtenir le champ de vent autour de la Corse. Ce territoire m'a semblé assez pertinent à étudier. On y trouve du relief et la Méditerranée ce qui permet d'obtenir des vents assez forts sur certaines parties de l'île (*voir partie Climat* <https://fr.wikipedia.org/wiki/Corse>).

Le code discuté ci-dessous est fourni. Je vais expliquer les parties les plus importantes du code permettant d'afficher la carte des vents de la Corse.

## 2.1 Explication du code

1. **Choix du maillage.** La zone étudiée sera de 90km de largeur et de 200km de longueur. On va diviser cette zone en 11 segments pour la largeur (soit un pas de ~8km) et en 16 segments pour la longueur (soit un pas de 12.5km). On va donc de 41.5° à 43° par pas de 0.1° pour la latitude et de 8.5° à 9.5° par pas de 0.1. On fera donc la requête de 176 vitesses et 176 directions de vent. Voici le code associé à cette partie.

```
latitudes = np.arange(41.5, 43.02, 0.1)
longitudes = np.arange(8.5, 9.52, 0.1)
```

2. **Création d'une liste de coordonnées.** Pour cette partie, j'ai simplement utilisé les fonctions repeat et tile de la bibliothèque numpy. Celles-ci me permettant d'obtenir les 176 couples (latitude, longitude) pour obtenir le maillage souhaité.

```
l_latitudes = np.repeat(latitudes_list, len(longitudes)).tolist()
l_longitudes = np.tile(longitudes_list, len(latitudes)).tolist()
```

3. **Utilisation de l'API et obtention du jeu de données.** L'API étant très bien documenté, j'ai pu facilement utiliser celle-ci pour obtenir mon jeu de données. Une partie du code étant fournie sur le site, j'ai simplement dû modifier les champs pour la latitude et la longitude afin de donner la liste de latitudes & longitudes. J'obtiens les vitesses et les directions des vents sur chaque point pour chaque heure de la journée du 09/09/2024 (choisi arbitrairement).

```
import requests_cache
import pandas as pd
from retry_requests import retry
# Setup the Open-Meteo API client with cache and
#retry on error cache_session = requests_cache.CachedSession('.cache', expire_after = -1)
retry_session = retry(cache_session, retries = 5, backoff_factor = 0.2)
openmeteo = openmeteo_requests.Client(session = retry_session)
# Make sure all required weather variables are listed here
# The order of variables in hourly or daily is important to assign them correctly below
url = "https://archive-api.open-meteo.com/v1/archive"
params = {
    "latitude": l_latitudes,
    "longitude": l_longitudes,
    "start_date": "2024-09-09",
    "end_date": "2024-09-09",
    "hourly": ["wind_speed_10m", "wind_direction_10m"] }
responses = openmeteo.weather_api(url, params=params)
```

4. **Extraction des données à 0h00.** Après avoir obtenu toutes les données, j'ai décidé d'essayer d'afficher le champ de vent pour la première heure du jeu de données soit 0h00. Pour chaque couple (latitude, longitude), j'ai extrait la direction et la vitesse du vent. Je suis ensuite passé des coordonnées polaires aux coordonnées cartésiennes en convertissant la vitesse et la direction en coordonnées d'un vecteur de cette manière :

```
wind_direction_radians = np.deg2rad(wind_dir_tmp)
wind_data.at[i, 'u'] = wind_spd_tmp * np.cos(wind_direction_radians)
wind_data.at[i, 'v'] = wind_spd_tmp * np.sin(wind_direction_radians)
```

5. **Création de la carte.** Enfin, après avoir obtenu nos données, on les affiche en utilisant les librairies cartopy et matplotlib. J'affiche les vecteurs sous la forme de quiver (flèche représentant les vecteurs). J'obtiens la figure di-dessous via le code suivant :

```

# Création de la carte
plt.figure(figsize=(10, 8))
ax = plt.axes(projection=ccrs.Mercator())
# Tracer les contours de la Corse
ax.coastlines()
ax.set_extent([8.25, 9.6, 41.2, 43.1], crs=ccrs.PlateCarree())
ax.set_title("Direction et vitesse du vent à l'heure 0")
ax.quiver(wind_data['lon'], wind_data['lat'], wind_data['u'], wind_data['v'], scale=100,
transform=ccrs.PlateCarree(), color='r')
plt.show()

```

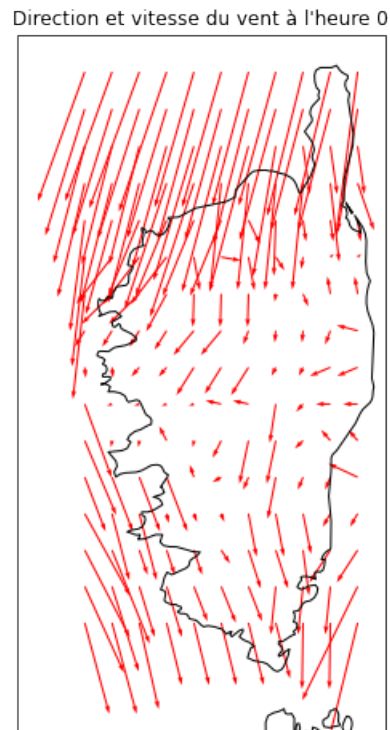


FIGURE 3 –  
Champ de vents en Corse le 09/09/2024 à 0h00.