Optimisation de l'étude de mise en place d'une éolienne

Enjeux sociétaux:environnement,sécurité,énergie

N°candidat:49001

Introduction

- L'énergie éolienne est une énergie renouvelable dont le potentiel théorique mondial est de 10^6 TWh/an, La part de l'éolien dans la production mondiale d'électricité atteignait 4,8 % en 2018 et est estimée à 5,3 % en 2019.c'est pour cela qu'il est de grande importance pour les êtres humains d'améliorer leurs moyens pour pouvoir profiter de cette richesse.

Plan

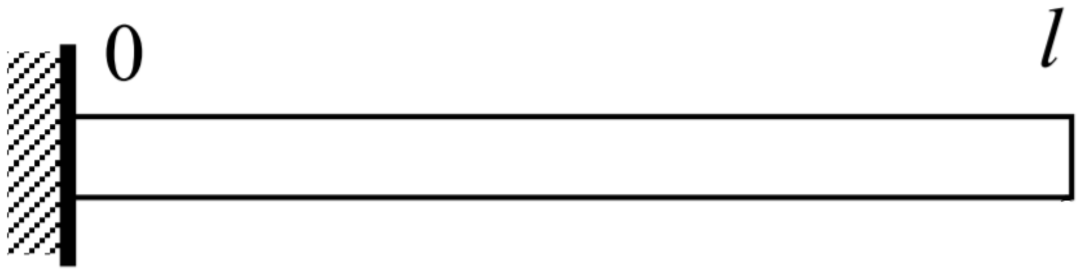
- Détermination d'une modélisation qui permet d'étudier les fréquences propres de flexion d'une pale éolienne
- Résolution de l'équation par des méthodes analytiques en utilisant python.
- Présentation de la méthode SWEPT pour faire l'étude préliminaire de la mise en place d'une turbine
- Conclusion.

1_Etude mécanique d'une pale éolienne

- Les pales éoliennes sont de plus en plus longues, et elles sont fabriquées à partir de matériaux plus légers et plus flexibles,cela est principalement fait pour augmenter leur production énergitique, ce qui les rend sujettes aux actions du vent et aux tremblements.

1-1:Hypothèses

- On modélise une pale éolienne par une poutre de longueur L, de surface de section A, encastrée au moyeux (x=0) et libre à l'extrémité (x=L) , cela nous permet d'étudier la déformation en flexion d'une telle structure .
- Les ondes qui résultent en une déformation de torsion ou une vibration longitudinale seront négligées .

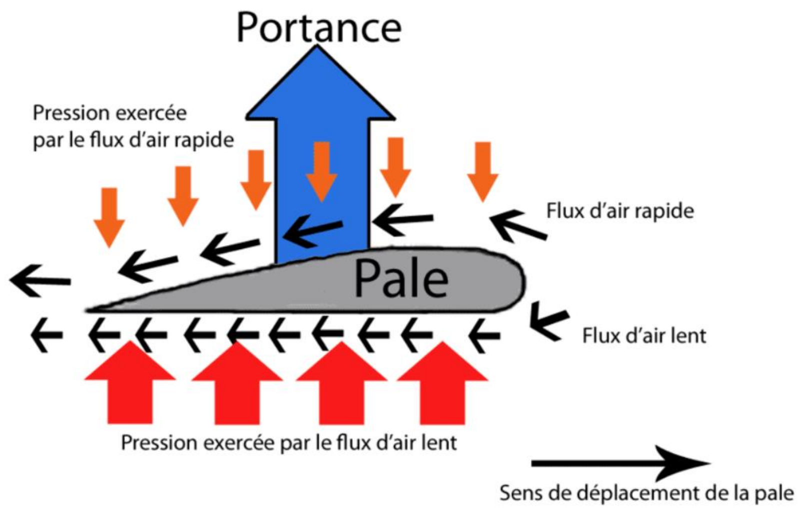


1-2: Actions aérodynamiques sur une pale

- La Force de portance:

$$F_z = \frac{1}{2}\rho V^2 S C_z$$

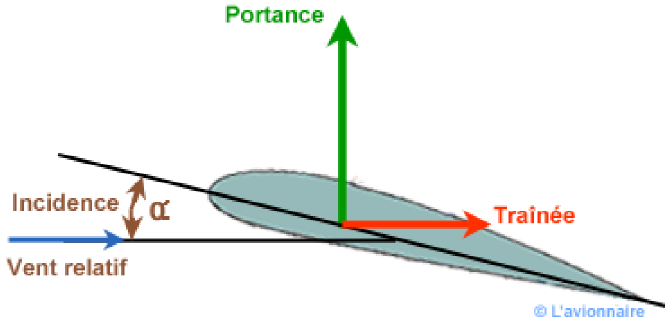
- ρ : Masse volumique de l'air(kg/m3)
- V : vitesse en m/s
- S : surface de référence de la pale en m2
- C_z = coefficient de portance (nombre sans dimension)



- La trainée

$$F_x = \frac{1}{2}\rho S C_x V^2$$

- ρ : Masse volumique de l'air
- V : Vitesse de déplacement en m/s
- S : Surface de référence (projetée)
- C_z Coefficient du trainée



1-3: Etude de la vibration de flexion d'une pale

$$\boxed{\frac{EI}{\rho A}\frac{\partial^4 w(x,t)}{\partial x^4}+\frac{\partial^2 w(x,t)}{\partial t^2}=0}$$

- W(x,t)=déplacement
- E=module d'élasticité du matériau
- A=surface de la section
- ρ=masse volumique
- I=moment d'inertie de la section

Cette équation est basée sur la théorie des poutres **d'Euler-Bernouilli** qui fait deux approximations importantes:

- Les déformations de la section droite dues au cisaillement sont négligées.
- L'effet d'inertie de rotation est négligé

Séparation des variables

W(x,t)=X(x)T(t)

$$\mu^2 \frac{X^{(4)}(x)}{X(x)} = -\frac{\ddot{T}(t)}{T(t)} = \omega^2 \qquad \text{avec} \qquad \mu = \sqrt{\frac{EI}{\rho A}}$$

On prend cst=-w² car si cst>0 alors T(t) diverge

$$\left\{ \begin{array}{l} \dfrac{\partial^2 T}{\partial t^2} + \omega^2 T(t) = 0 \quad \Rightarrow \quad T(t) = a\sin(\omega t) + b\cos(\omega t) \\[2em] X^{(4)}(x) - \beta^4 X(x) = 0 \qquad \text{avec} \quad \beta^4 = \dfrac{\omega^2}{\mu^2} = \dfrac{\rho A \omega^2}{EI} \end{array} \right.$$

Equation caractéristique: $r^4 = \beta^4 \Rightarrow X(x) = D_1 e^{\beta} + D_2 e^{-\beta} + D_3 e^{j\beta} + D_4 e^{-j\beta}$

1-4: Conditions au limites

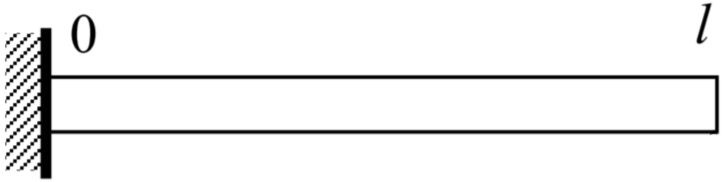
- Déplacement du à la flexion nulle à l'extrémité encastrée
- Déplacement angulaire(rotation) du à la flexion nulle à l'extrémité encastrée
- Moment de flexion nulle à l'extremité libre
- Force de cisaillement nulle à l'extremité libre

extrémité **libre**  x_0	pas de contraintes sur le déplacement et la rotation $\Rightarrow$ le **moment de flexion** et la **force de cisaillement** s'annulent en x_0	$EI\left.\dfrac{\partial^2 w(x,t)}{\partial x^2}\right\|_{x=x_0}=0$ $-EI\left.\dfrac{\partial^3 w(x,t)}{\partial x^3}\right\|_{x=x_0}=0$
extrémité **encastrée**  x_0	les **déplacements** transversal et angulaires sont **nuls** en x_0	$w(x_0,t)=0$ $\theta(x_0,t)=\left.\dfrac{\partial w(x,t)}{\partial x}\right\|_{x=x_0}=0$

Détermination des constantes

- X(x) peut ainsi s'écrire: $X(x) = C_1 sin(\beta x) + C_2 cos(\beta x) + C_3 sinh(\beta x) + C_4 cosh(\beta x)$

$$\begin{cases} X(0) = 0 & \Rightarrow C2 + C4 = 0 \\ X'(0) = 0 & \Rightarrow \beta(C1 + C3) = 0 \end{cases}$$



$$\begin{cases} X''(l) = 0 & \Rightarrow \beta^2 \left[-C_1 sin(\beta l) - C_2 cos(\beta l) + C_3 sinh(\beta l) + C_4 cosh(\beta l)\right] = 0 \\ X'''(l) = 0 & \Rightarrow \beta^3 \left[-C_1 cos(\beta l) + C_2 sin(\beta l) + C_3 cosh(\beta l) + C_4 sinh(\beta l)\right] = 0 \end{cases}$$

On obtient: C2 = −C4 et C1 = −C3 .

1-5:Equation des vibrations

$$\begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ -sin(\beta l) & -cos(\beta l) & sinh(\beta l) & cosh(\beta l) \\ -cos(\beta l) & sin(\beta l) & cosh(\beta l) & sinh(\beta l) \end{pmatrix} \begin{pmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

La solution (vecteur [C1 C2 C3 C4]T non-nul) s'obtient en trouvant la solution de l'équation

det(A) = 0 , soit: **det(A)=2cosh(βL)cos(βL)+2=0**

Alors les fréquences propres de la pale sont donnée par les racines de cette équation:

$$cosh(\beta_i)cos(\beta_i) + 1 = 0 \quad \text{(E1)} \qquad \beta_i^4 = 4\Pi^2 f_i^2 \frac{mL^3}{EI}$$

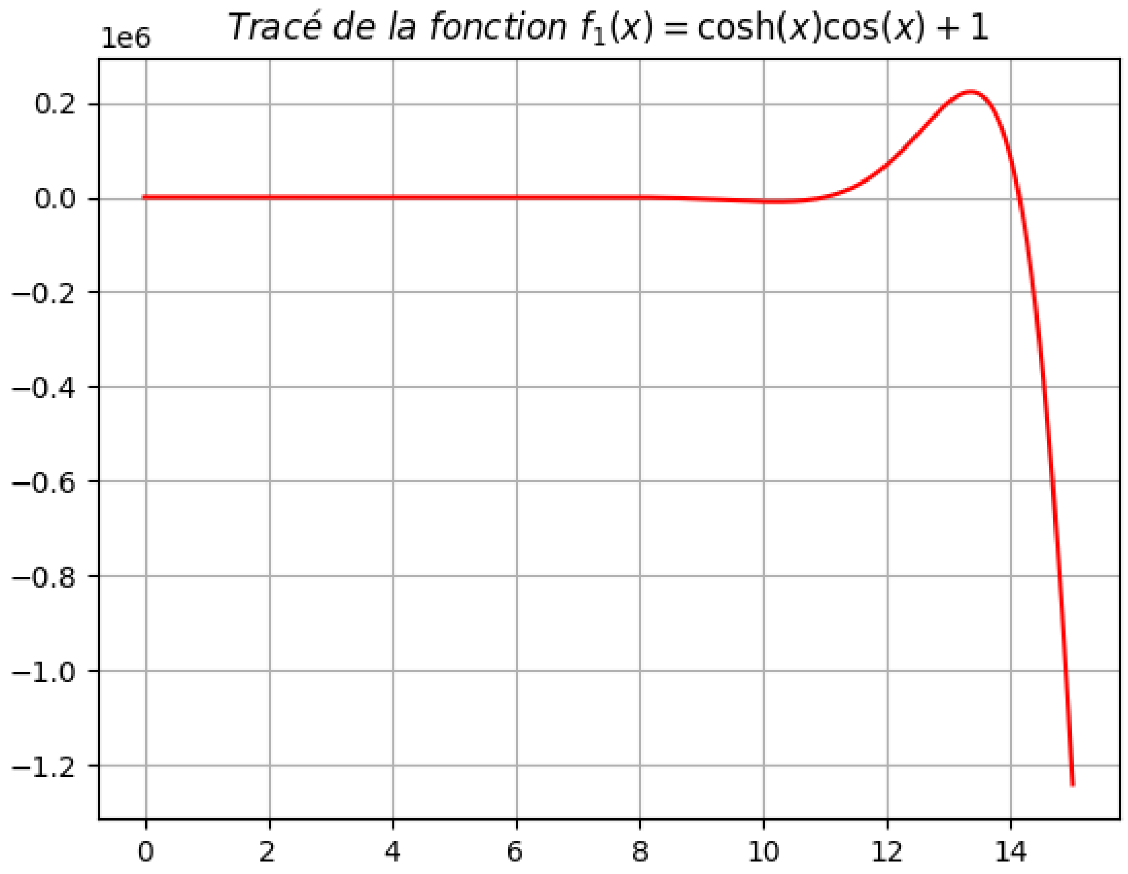
avec : - m , masse de la pale,
- L , longueur de la pale,
- E , module d'élasticité du matériau,
- I , moment quadratique de la section.

1-6: Recherche de solutions

- Afin de se rendre compte de la position, du nombre et des valeurs des racines de l'équation (E1), on propose d'en obtenir un tracé sur l'intervalle β ∈[0,15] .(voir annexe)

- On constate qu'il est difficile d'estimer le nombre (et la valeur) des racines dans l'intervalle demandé, ce qui était prévisible puisque lorsque

β >> 1, cosh (β) ≈ $\frac{1}{2}e^{\beta}$, la valeur de la fonction devenant alors **très grande !**



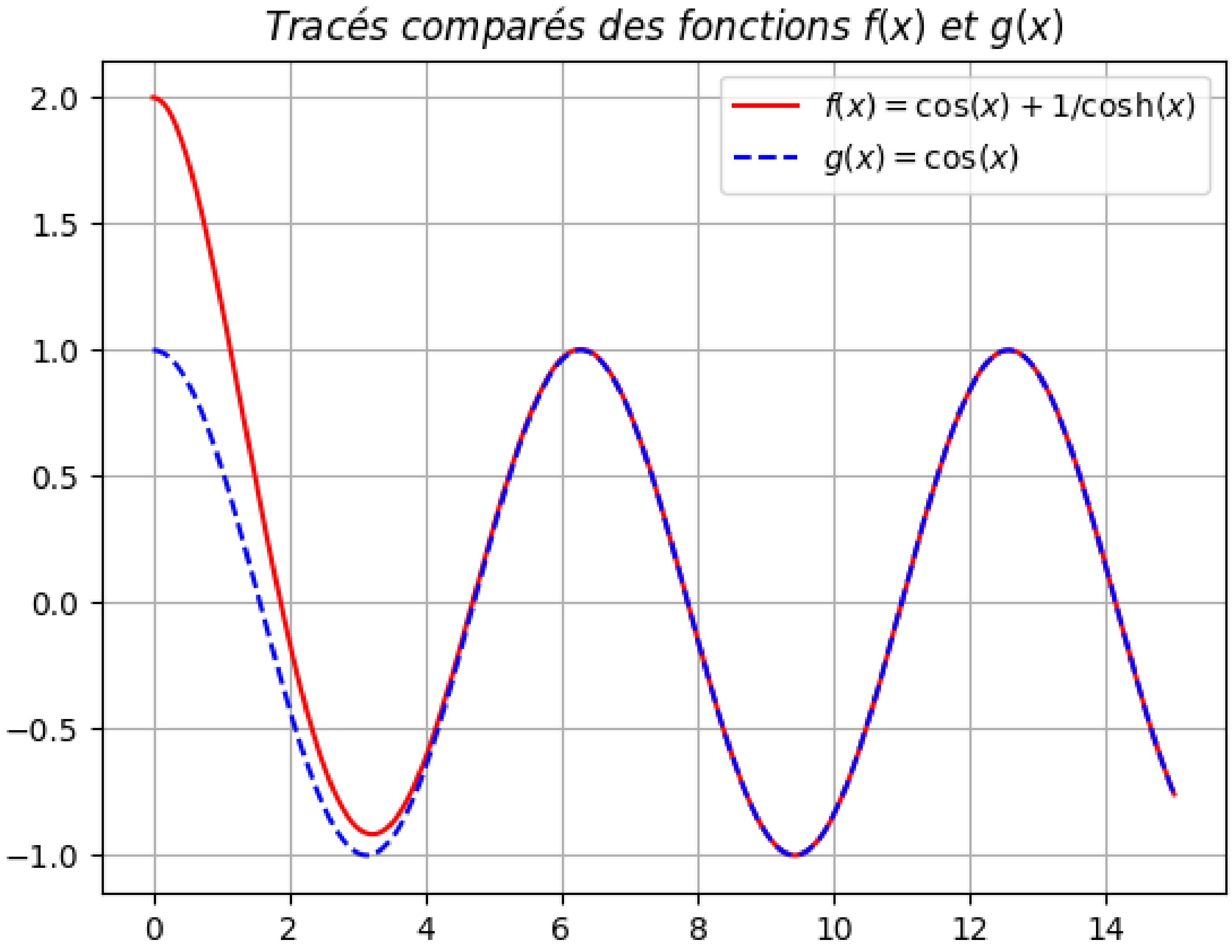
Equation équivalente

Pour mieux visualiser les racines on peut faire apparaitre une équation équivalente qui admet **les même racines que (E1)**:

$$cos(\beta) + \frac{1}{cosh(\beta)} = 0$$

Cette équation peut être approximée pour les grandes valeurs(n>5) par:

$$\beta_n = (2n+1)\frac{\pi}{2}$$



Méthodes numériques

- Pour déterminer les premières racines de l'équation on fait l'exécution de notre programme sur les intervalles convenables.
- Méthode de dichotomie:

  (voir annexe)

Principe de fonctionnement:

$$x_{moy} = \frac{g+d}{2}$$

$$si : f(g) * f(x_{moy}) > 0 \Rightarrow \quad g \leftarrow x_{moy}$$
$$sinon : \quad d \leftarrow x_{moy}$$

- $f(x)$ *la fonction dont un cherche une racine,*
- (g,d) *l'intervalle de départ dans lequel on cherche une racine,*
- $\varepsilon(=\mathrm{eps})$ *la précision souhaitée (critère d'arrêt).*

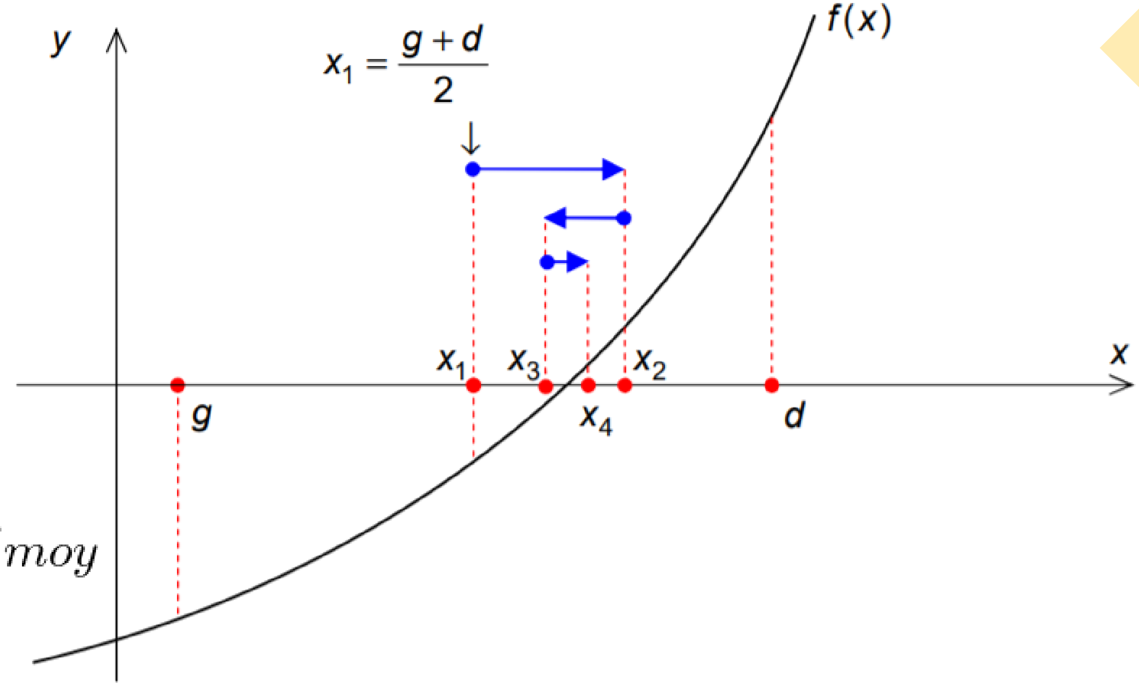


Illustration graphique du principe de la méthode de dichotomie.

Trace d'exécution

```
intervalle de recherche :  (4.0, 5.0) précision souhaitée =  1e-08
Méthode : dichotomie
racine =  4.694091133773327 Nb ditérations =  27 TempsCPU (ms) = 0.227
None
```

- On constate que cette méthode peut fournir des résultats assez précis (10^{-8}) .
- Mais ,elle nécessite un **grand nombre d'itérations** pour trouver ce résultat(27).
- On cherche une méthode moins coûteuse...

Méthodes numériques

- $f(x)$ la fonction dont un cherche une racine, et $fd(x)$, sa fonction dérivée (à l'ordre 1),
- x_0, la valeur de départ pour la recherche d'une racine,
- $\varepsilon(=\texttt{eps})$ la précision souhaitée (critère d'arrêt).

- Méthode de newton:

1. En utilisant la fonction dérivée
2. Sans utiliser la dérivée

$$U_0 \in [g, d] \;\; et \;\; U_{n+1} = U_n - \frac{f(U_n)}{f'(U_n)}$$

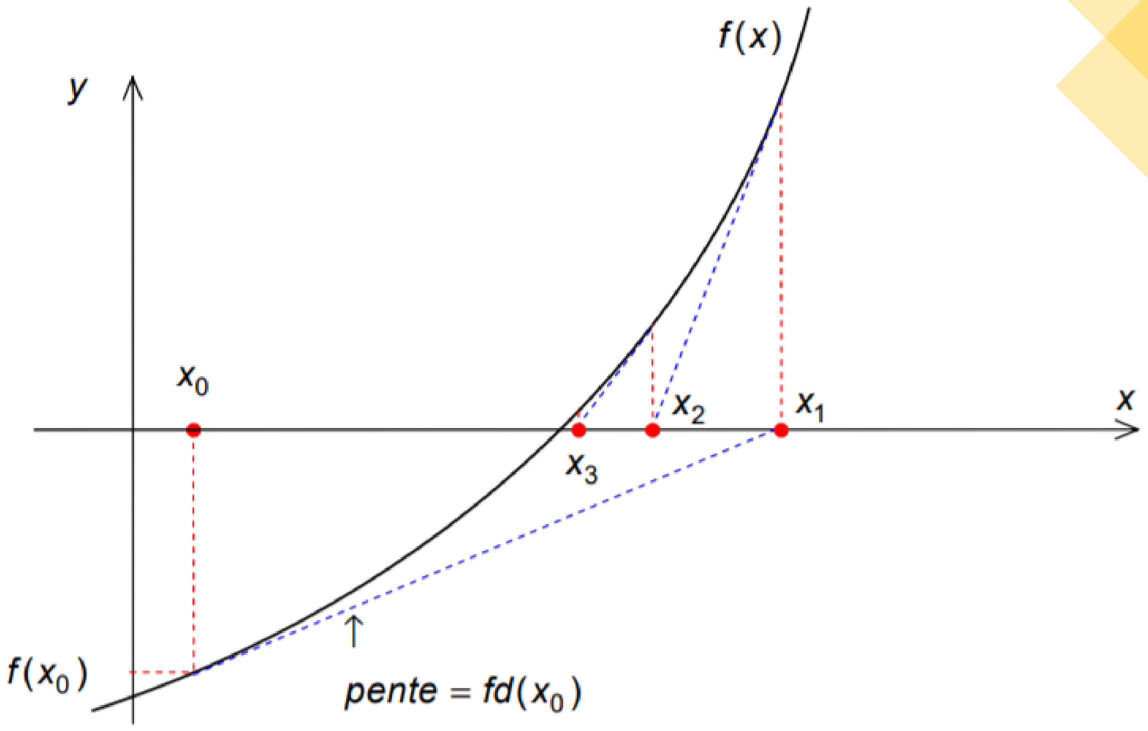


Illustration graphique du principe de la méthode de Newton.

Trace d'exécution

```
Méthode : Newton (avec dérivée)
racine =  4.694091132974175 Nb ditérations =  4 TempsCPU (ms) = 0.144
None
```

- La méthode de newton retrouve le même résultat avec seulement 4 itérations ce qui est bien plus efficace que l'autre méthode .
- En plus , pour **chaque itération calculée la précision se double** avec cette méthode:

$$soit\ k\ tel\ que\ \ U_1 - \beta_{n0} \leq k \qquad U_n - \beta_{n0} \leq 2\beta_{n0}\left(\frac{k}{2\beta_{n0}}\right)^{2n-1}$$

La méthode de newton nécessite la connaissance de la fonction dérivée qui n'est pas toujours facile à déterminer . ⇨ il est possible de faire une approximation de la dérivée.

Trace d'exécution

```
Méthode : Newton (variante sans dérivée)
racine =  4.694091132974174 Nb ditérations =  4 TempsCPU (ms) = 0.178
None
```

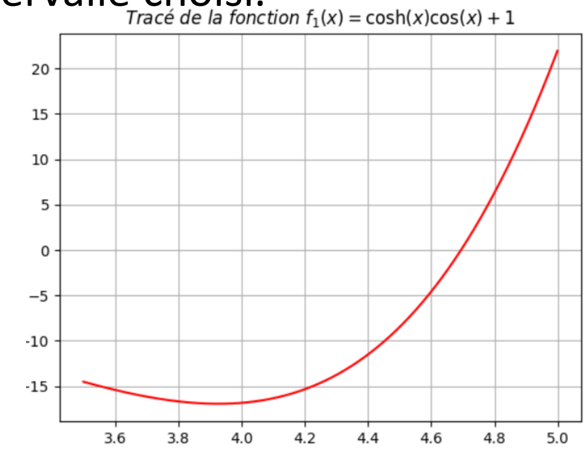
- Cette fonction prend en argument la fonction, le point de départ et la précision seulement.
- La fonction dérivée est approximée avec:

On prend h=e(précision)

$$fd(x) = \frac{f(x+h) - f(x-h)}{2h} \quad , \; h << 1$$

Alors qu'elle se caractérise par la rapidité et la précision, la méthode de newton risque de donner des résultats complètement faux si le bon choix de l'intervalle n'est pas fait:
Dans notre cas il faut que la fonction dérivée ne s'annule jamais dans l'intervalle choisi.

```
intervalle de recherche :  (3.5, 5.0) précision souhaitée =  1e-08
Méthode : dichotomie
racine =  4.694091133773327 Nb ditérations =  28 TempsCPU (ms) = 0.244
None
Méthode : Newton (avec dérivée)
racine =  10.995540734875467 Nb ditérations =  5 TempsCPU (ms) = 0.178
None
Méthode : Newton (variante sans dérivée)
racine =  10.99554073487545 Nb ditérations =  5 TempsCPU (ms) = 0.219
None
```



1-7: Résultat final

```
racine =  1.875104069709778 Nb ditérations =  27 TempsCPU (ms) = 0.165
None
racine =  4.694091133773327 Nb ditérations =  27 TempsCPU (ms) = 0.159
None
racine =  7.854757443070412 Nb ditérations =  27 TempsCPU (ms) = 0.159
None
racine =  10.995540738105774 Nb ditérations =  28 TempsCPU (ms) = 0.165
None
```

- Les pales subissent alors des vibrations . Ceci peut avoir un impact négatif sur leurs performances et peut même entraîner des pannes prématurées. De tels défauts peuvent faire obstacle à l’adoption des éoliennes pour la production d’électricité, en particulier sur les installations au large (offshore) qui sont difficiles d’accès(exemple: Siemens Gamesa « SG 14 - 222 DD »d'un rotor de **222** mètres de diamètre.

Les pulsation propres se calculent :

$$\omega_n = (\beta_n l)^2 \sqrt{EI / \rho A l^4}$$

2: Estimation du potentiel éolien

- En pratique l'étude de mise en place d'une éolienne nécessite beaucoup de ressources financières et même technologiques pour pouvoir estimer précisément le potentiel éolien d'un terrain avec une turbine donnée .
- C'est pourquoi il est intéressant d'avoir une manière de juger l'ordre de grandeur de ce potentiel avant de commencer les travaux de recherche.

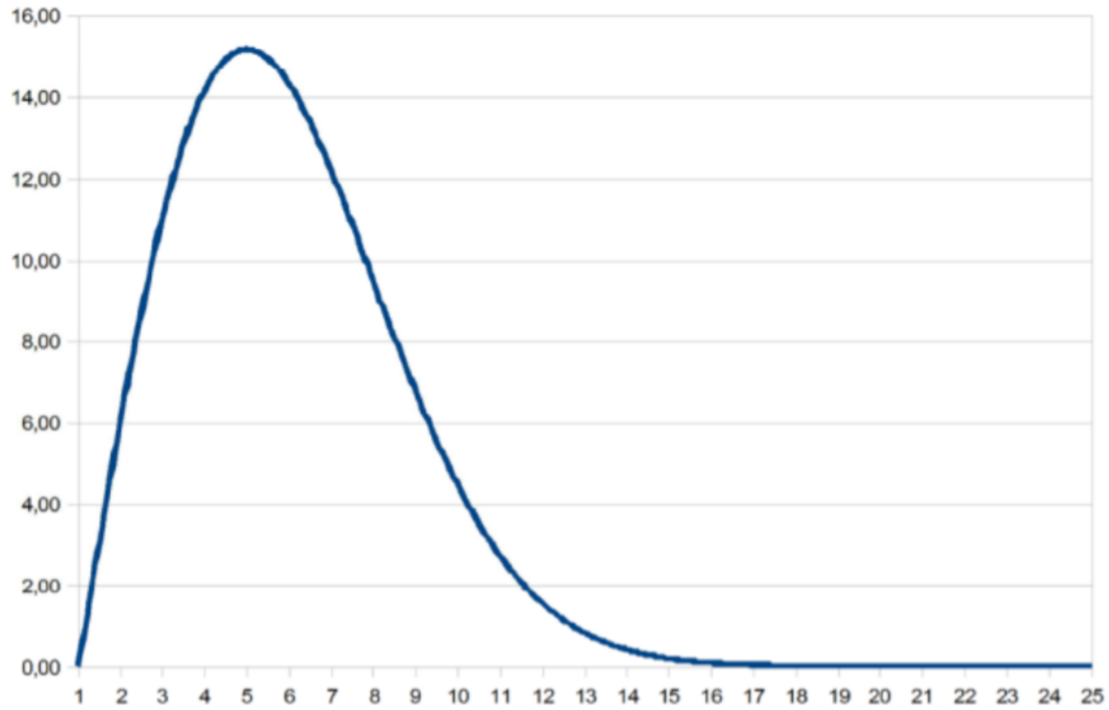
Une approche simplifiée pour l'évaluation du potentiel éolien: la méthodologie SWEPT

- Cette approche peut être décrite en quelques étapes basées sur trois caractéristiques générales:

1. l'estimation du vent (WE) doit être effectuée; c'est-à-dire que la répartition du vent est connue;
2. le lieu (P) doit être connu; c'est-à-dire que la géographie locale et la surface de rugosité sont connues;
3. la turbine (T) est complètement définie; c'est-à-dire que ses caractéristiques sont connues.

2-1: L'étape WE: estimation de la distribution de la vitesse du vent.

- La première étape consiste en le calcul de la courbe de répartition du vent. Si des données mesurées sont disponibles,
- la courbe de distribution sera déduite des données en utilisant une **distribution de Rayleigh**, ce qui est un cas particulier de la distribution de Weibull à deux paramètres.



G. 2 – Rayleigh distribution for a given value of V_m

- Cependant, dans certains cas, la vitesse moyenne n'est pas disponible car cela suppose qu'au moins 18 mois de mesures ont été effectueés sur le lieu où une éolienne doit être installée.
- Dans ce cas, une alternative est d'estimer cette quantité si elle est connue à un endroit non loin du site choisi.

- Cette méthode propose d'utiliser une donnée subjective: la plus «Occurrente» vitesse (MOV). Cette méthode consiste en une estimation de la MOV, à l'aide d'une échelle dérivée du "**Beaufort scale"**.
- Le tableau représente une transposition de la vitesse du vent à partir d'éléments subjectifs.
- Bien sûr, la plupart de sites éoliens ont un nombre de Beaufort inférieur à 6.

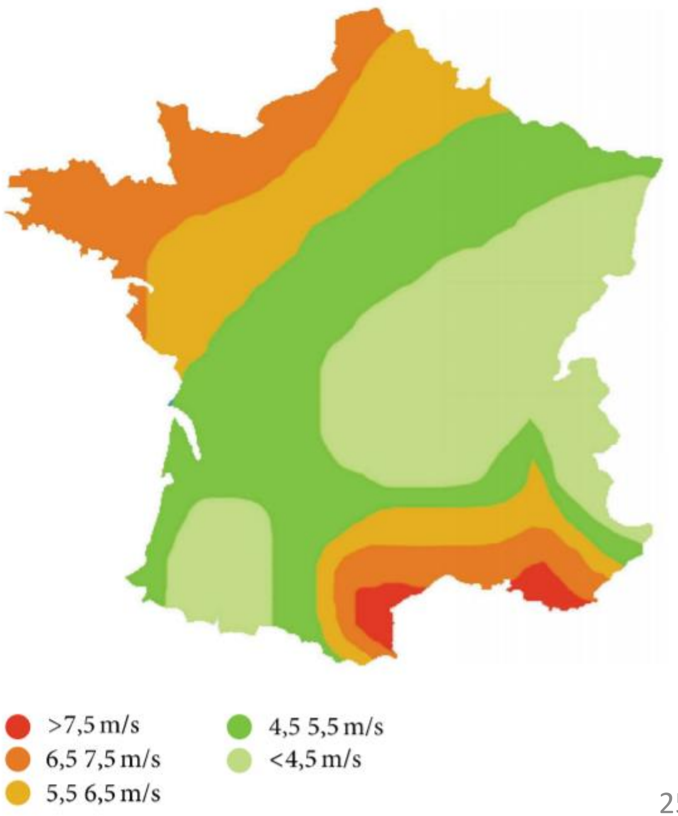
Beaufort number	Estimated Velocity (m/s)	International description	Observed conditions
0	<1	calm	Smoke rises vertically
1	1	Light air	Direction of wind shown by smoke drift but not by wind vanes
2	2	Light brise	Wind felt on face: leaves rustle, vanes move by wind
3	4	Gentle breeze	Leaves and small twigs in constant motion; wind extends light flag
4	7	Moderate	Raises duct, loose paper; small branches move
5	10	Fresh	Small trees in leaf begin to sway; crested wavelets form on inland waters
6	12	Strong	Large branches in motion; whistling heard in telephone wires; umbrellas used with difficulty
7	15	Near gale	Whole trees in motion; resistance felt walking against wind
8	18	gale	Breaks twigsoff trees; impedes walking
9	20	Strong gale	Slight structural damages occurs
10	26	Storm	Trees uprooted; considerable damage
11	30	Violent storm	Widespread damage

TABLE – on-shore Beaufort scale

Quelques exemples numériques:

Vitesse du vent en m/s	8	10	12	14	16	18	20	22
Nombre de beaufort	4	5	6	7	7	8	8	9

Plage de vitesse moyenne estimée à 10m de hauteur (France):

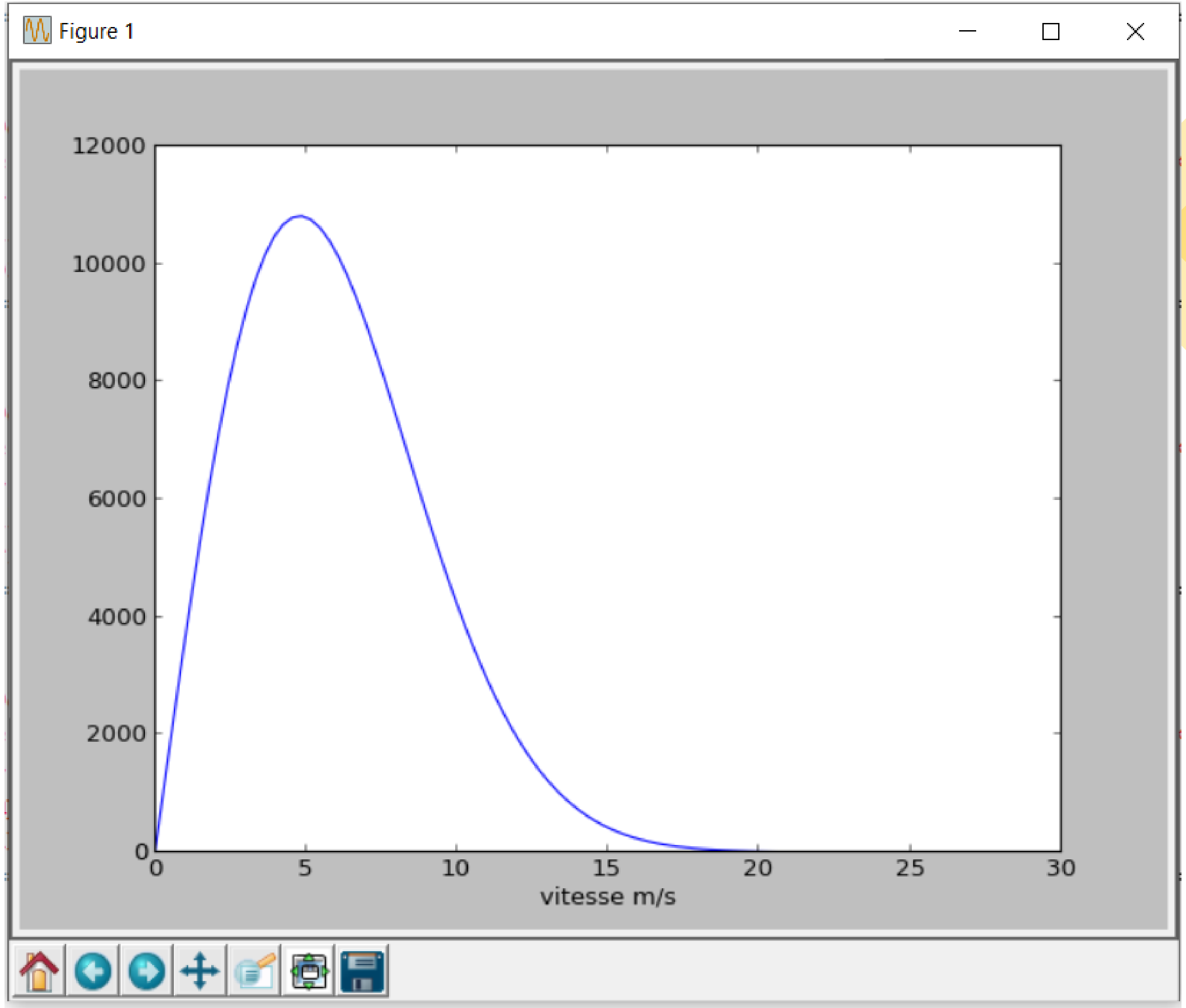


$$V_m = \frac{MOV \, . \, \sqrt{2\,\pi}}{2}$$

Pour une distribution de Rayleigh, la vitesse moyenne peut être facilement exprimée à partir de la MOV selon l'équation suivante :

Génération de la distribution du vent avec python

Cette distribution est générée avec une fonction python qui prend en argument la vitesse moyenne seulement(voir annexe):



2-2: L'étape P :configuration du site

- Dans cette deuxième étape, la répartition précédente est adaptée en raison de la hauteur supposée de l'éolienne.
- Une loi de puissance classique est proposée, et directement liée à la hauteur h de la mât .

- La loi de **Davenport et Harris** est une loi empirique qui relie la vitesse du vent à une hauteur connue à sa vitesse à une deuxième hauteur , En effet la vitesse du vent à la hauteur où se trouve le moyeu du rotor principal de la turbine est déduite à partir de la vitesse du vent « mesurée » V à la hauteur h.

$$\frac{V_T}{V} = \left(\frac{h_T}{h}\right)^{\alpha}$$

- V_T : vitesse du vent a la hauteur h_T du turbine
- (alpha):coéfficient decrivant l'état du terrain

- La valeur commune du paramètre(alpha)est souvent prise égale à 1/7.

Bien entendu, la valeur dépend de la configuration du terrain et la rugosité.

- Ensuite, la distribution du vent peut être estimée à la hauteur du moyeu du rotor principal.

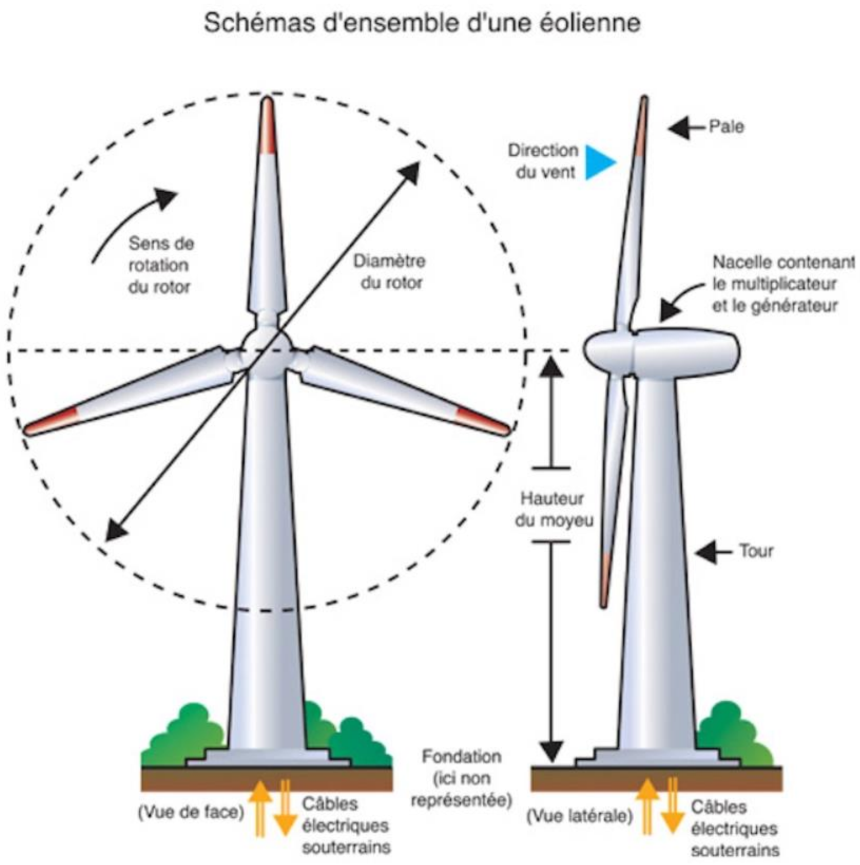
Ground type	α
Ice	0.07
Snow on a flat ground	0.09
Calm sea	0.09
Short cut grass	0.14
Meadow	0.16
Cereal cultures	0.19
Hurdles	0.21
Trees and scattered hurdles	0.24
Trees and dense hurdles	0.29
Sub-urban or urban site	0.31
Forest	0.43

TABLE – estimation of the velocity from the ground roughness knowledge

Ces calculs conduisent à la distribution estimée du vent à la hauteur souhaitée de l'éolienne.
Laissez-nous rappeler que seule la « connaissance subjective » du vent et de la hauteur du moyeu du rotor principal sont nécessaires pour obtenir se résultat.

2-3: L'étape T : choix de la turbine.

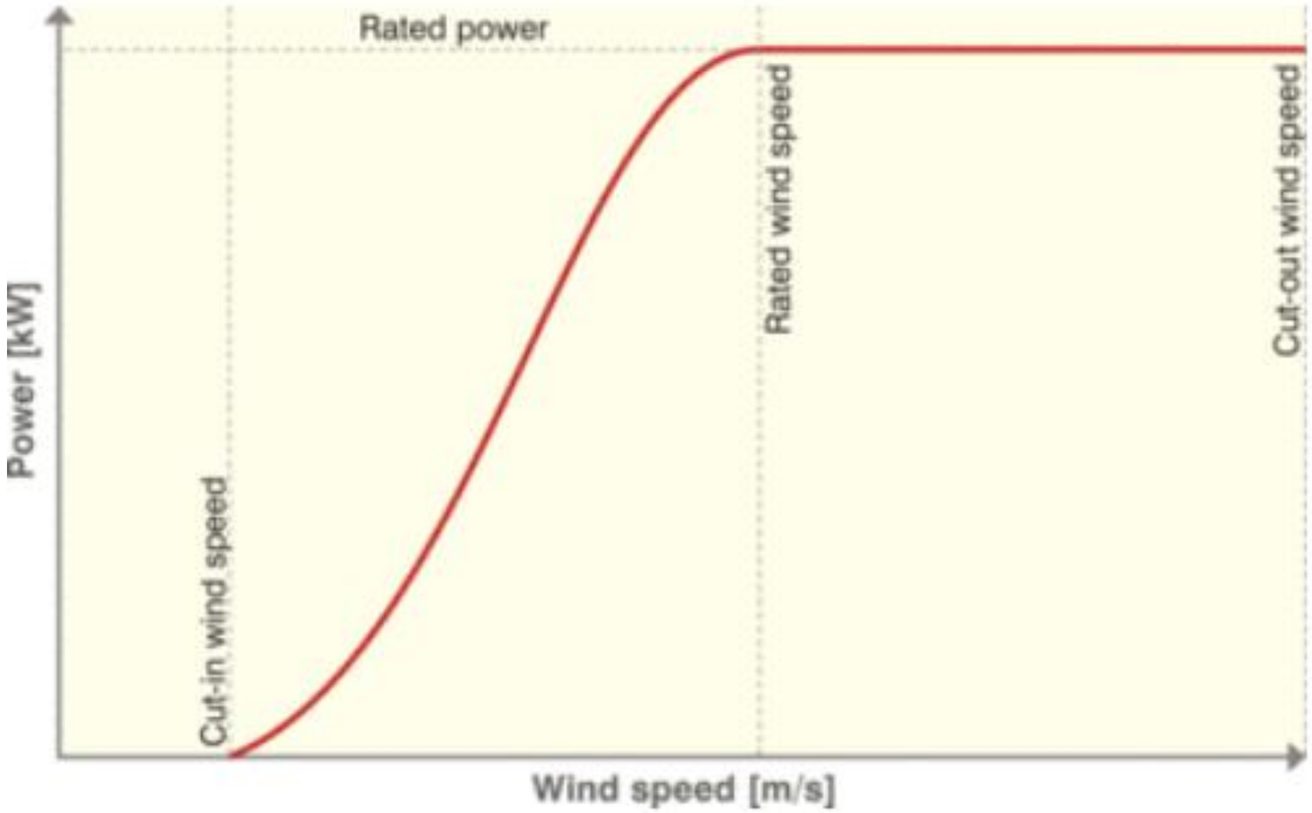
- les éoliennes sont généralement choisies avant le calcul de l'énergie livrée, et ne sont pas
- directement inclus dans la méthodologie.
- En fait, les installateurs ont généralement une idée du rotor et ils utilisent « juste » la courbe électrique (puissance électrique vs vitesse du vent) pour calculer la production d'énergie à partir de la distribution des vitesses sur un site donné.



- La **<u>courbe caractéristique</u>** de puissance d'une éolienne donne la puissance électrique en fonction de la vitesse du vent. Généralement, ces courbes sont données par les fabricants d'éoliennes.

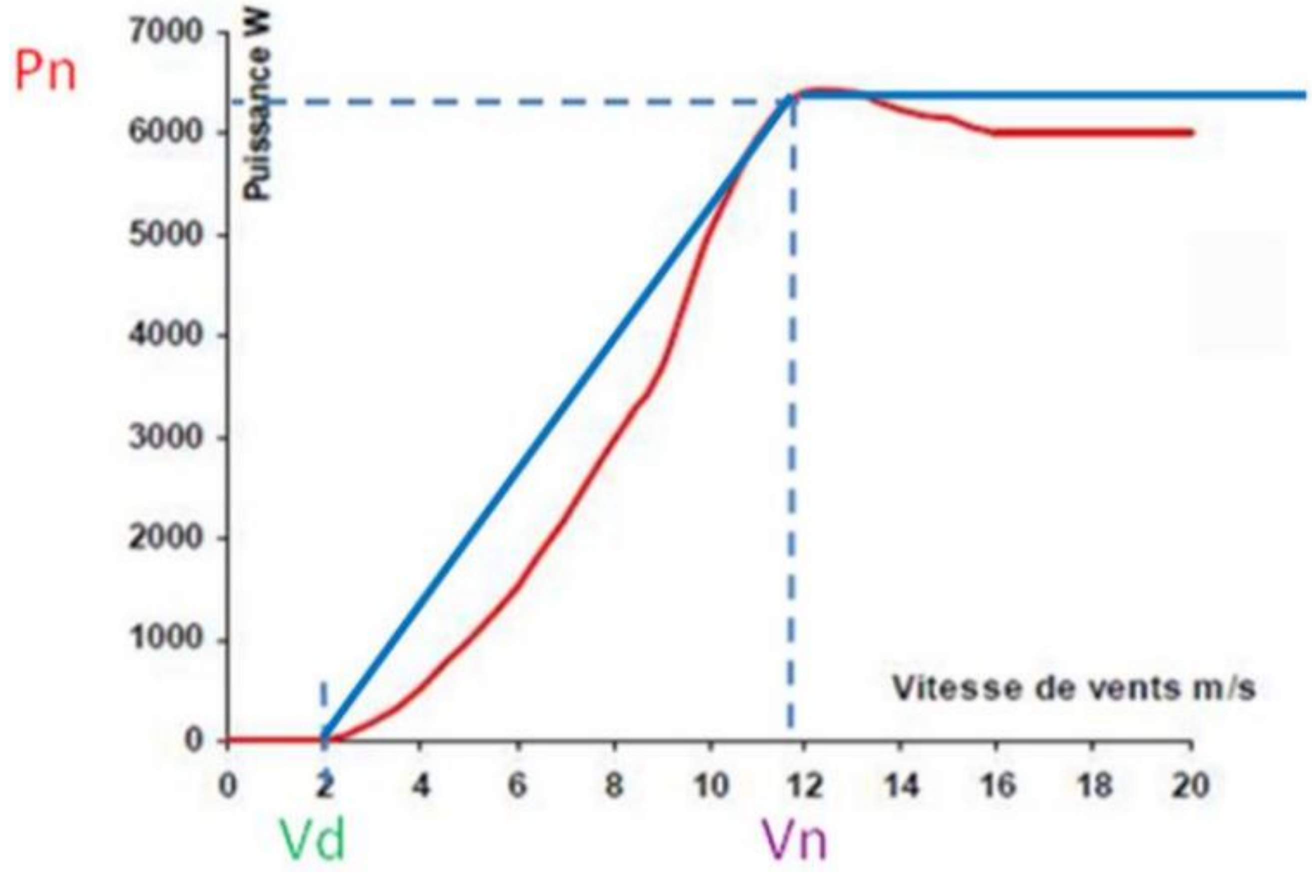
La courbe caractéristique de puissance comporte trois grands paramètres :

- <u>La vitesse minimale de démarrage (*cut-in wind speed*) :</u> il s'agit de la vitesse du vent à partir de laquelle l'éolienne commence à débiter une puissance utile (c'est-à-dire de la puissance électrique).
- <u>La vitesse maximale ou d'arrêt (*cut-off wind speed*) :</u> il s'agit de la vitesse maximale acceptable par l'éolienne. Au-delà de celle-ci, Si le vent présente une vitesse supérieure, l'éolienne est mise à l'arrêt.
- <u>La puissance nominale (*rated power*) :</u> cette valeur est souvent égale à la puissance électrique maximale qui peut être extraite de l'éolienne.



Approximation de la courbe caractéristique

- Le point d'arrêt est situé pour des vitesses élevées qui sont très rares, de sorte que l'énergie correspondante au cours d'une année est très faible.
- Sous ces considérations, la courbe de puissance de l'éolienne peut facilement être approximée par deux lignes, ce qui facilite le calcul de la production d'électricité au regard de la distribution, pour chaque valeur de la vitesse, le nombre d'heures de vent et la puissance délivrée sont connus, et l'énergie n'est que le produit de la puissance par le nombre d'heures à cette puissance.



2-4 : AUTRES CONSIDERATIONS: Nombre total des éoliennes à placer dans le site

- I = Dimension du terrain perpendiculaire à la direction prédominante du vent
- L = Dimension du terrain parallèlement à la direction prédominante du vent
- D = Diamètre du rotor de la machine
- H = Hauteur du pylône
- N1= Nombre d'aérogénérateurs par rangée
- N2 = Nombre de rangée d'aérogénérateurs
- N = Nombre total d'aérogénérateurs à placer sur le site.

- Conditions à respecter:

$$(N1 + 1) \times 10H < I$$

$$(N2 + 1) \times 3D < L$$

$$N = N2 \times N1$$

CONCLUSION

ANNEXE

tipe2.py (C:\Users\kallel_PC\Desktop\TIPE 2021\tipe2.py) - Interactive Editor for Python

Fichier Édition Affichage Paramètres Shell Exécuter Outils Aide

<tmp 1> | tipe2.py | tipe2021.py

```python
from math import *  # pour pouvoir manipuler 'pi'
import pylab as plt
import matplotlib.pyplot as plt
import numpy as np
import time

# définition de la fonction dont on cherche une racine et fonction approchée
def f1(x):
    return np.cos(x)*np.cosh(x)+1
def f(x):
    return np.cos(x) + 1/np.cosh(x)
def fd(x):
    return -np.sin(x) -  np.sinh(x)/np.cosh(x)**2
def g(x):
    return np.cos(x)

# tracé de la fonction sur un intervalle jugé pertinent pour prévisualisation
print("définition d'un intervalle pour prévisualisation de la fonction")
a = float(input("Entrer la borne inférieure de l'intervalle (a) : "))
b = float(input("Entrer la borne supérieure de l'intervalle (b) : "))
X = np.arange(a,b,0.001)

plt.figure(0)
plt.plot(X , f1(X) , '-r')
plt.title(r"$Tracé\ de\ la\ fonction\ f_1(x)=\cosh(x)\cos(x)+1$")
plt.grid(True)
plt.show()

plt.figure(1)
plt.plot(X , f(X) , '-r' , label=r"$f(x)=\cos(x)+1/\cosh(x)$")
plt.plot(X , g(X) , '--b', label=r"$g(x)=\cos(x)$")
plt.title(r"$Tracés\ comparés\ des\ fonctions\ f(x)\ et\ g(x)$")
plt.legend()
plt.grid(True)
plt.show(block=False)
plt.show()
```

ANNEXE

<tmp 1> tipe2.py tipe2021.py

```python

# construction d'une procédure de dichotomie dans l'intervalle (d,g)
def recherche_dicho (f , g , d , eps):
    tcpu0=time.perf_counter()
    j = 0
    if f(g)*f(d) >0 :
        return print("l'intervalle proposé ne permet pas d'appliquer",
                     "la méthode de la dichotomie !")
    while abs(g-d) > eps :
        j += 1
        if f(g)*f((d+g)/2) > 0:
            g = (d+g)/2
        else :
            d = (d+g)/2
    t=time.perf_counter()-tcpu0
    print('racine = ',d,'Nb d''itérations = ',j,'TempsCPU (ms) =',floor(1e6*t)/1000)
    return

# construction d'une procédure de Newton à partir de la valeur (x)
def recherche_newton (f , fd , x , e):
    tcpu0=time.perf_counter()
    j = 0
    while abs(f(x) / fd(x)) > e:
        j += 1
        if fd(x) ==0:
            return ("La méthode de Newton ne peut pas aboutir (dérivée nulle!)")
        x=x-f(x)/fd(x)
  #      print(x)
    t=time.perf_counter()-tcpu0
    print('racine = ',x,'Nb d''itérations = ',j,'TempsCPU (ms) =',floor(1e6*t)/1000)
    return

# construction d'une procédure de Newton (variante sans dérivée) à partir de (x)
def recherche_newton_variante (f , x , e):
    tcpu0=time.perf_counter()
    j = 0
    while abs( 2*f(x)/(f(x+e)-f(x-e)) ) > 1:
        j += 1
        if (f(x+e)-f(x-e)) == 0:
            return ("La méthode de Newton ne peut pas aboutir (dérivée nulle!)")
        x=x-2*e*f(x)/ (f(x+e)-f(x-e))
    t=time.perf_counter()-tcpu0
    print('racine = ',x,'Nb d''itérations = ',j,'TempsCPU (ms) =',floor(1e6*t)/1000)
    return
```

ANNEXE

```python

# application des méthodes de dichotomie et de Newton
g = float(input("Entrer la borne inférieure de l'intervalle (g) : "))
d = float(input("Entrer la borne supérieure de l'intervalle (d) : "))
e = 1e-8

print( "intervalle de recherche : ", (g , d) , "précision souhaitée = ", e )

print( "Méthode : dichotomie" )
print( recherche_dicho (f , g , d ,e) )
print( "Méthode : Newton (avec dérivée)" )
print( recherche_newton (f , fd , g ,e) )
print( "Méthode : Newton (variante sans dérivée)" )
print( recherche_newton_variante (f , g , e) )

import scipy as sp
import scipy.optimize
print( "Newton avec la bibliothèque scipy.optimize (avec dérivée):" )
tcpu0=time.perf_counter()
print(sp.optimize.newton(f,g,fprime=fd,tol=e))
print(time.perf_counter()-tcpu0)

print( "Newton avec la bibliothèque scipy.optimize (sans dérivée):" )
tcpu0=time.perf_counter()
print(sp.optimize.newton(f,g))
print(time.perf_counter()-tcpu0)

```

ANNEXE

tipe2021.py (C:\Users\kallel_PC\Desktop\TIPE 2021\tipe2021.py) - Interactive Editor for Python

Fichier Édition Affichage Paramètres Shell Exécuter Outils Aide

<tmp 1> tipe2.py tipe2021.py

```python
import matplotlib.pyplot as plt
import numpy as np
#generation de la distribution des vitesses du vent

def Rayleigh(v,vm,delta_v):
    return (delta_v*np.pi*v*np.exp(-(np.pi*(v/vm)**2)/4))/(2*vm**2)

x=np.linspace(0,30,100)
y=[]
for i in x:
    y.append(Rayleigh(i,6,1)*100)


plt.plot(x,y)
plt.xlabel("vitesse m/s")
plt.show()


```