



INSTITUT
NATIONAL
des SCIENCES
APPLIQUÉES



Projet de Physique P6-3
STPI/P6-3/2009 – n°40



Nicolas CHOCHOY Kok Choong WONG
Florent GANZENMULLER Xiaoyu XIE
Kévin RIBOT

Enseignant responsable du projet :

Ludovic Henriet

**Utilitaires de calcul de
ressources éoliennes**



À TAILLE
HUMAINE
À L'ECHELLE
DU MONDE

Date de remise du rapport : 22/06/09

Référence du projet : STPI/P6-3/2009 – n°40

Intitulé du projet : **Utilitaires de calcul de ressources éoliennes**

Type de projet : *Simulation, programmation*

Objectifs du projet:

- Comprendre le fonctionnement d'une éolienne.
- Implantation d'une éolienne sur un site à l'aide du logiciel Windpro.
- Etude des mesures de vent et calculs de productible à l'aide du logiciel Windpro.
- Création d'un utilitaire de calcul permettant de calculer la puissance fournie par une éolienne selon les différents secteurs de vent.
- Création de l'utilitaire de calcul sous JAVA

TABLE DES MATIERES

Introduction	5
I <u>METHODOLOGIE / ORGANISATION DU TRAVAIL</u>	6
1.1. Description de l'organisation adoptée pour le déroulement du travail.....	6
1.2. Organigramme des tâches réalisées et des étudiants concernés	7
II <u>TRAVAIL REALISE ET RESULTATS</u>	8
2.1. Windpro : Projet d'éolienne à Dieppe	8
2.1.1. Utiliser Windpro	8
2.1.2. Projet Dieppe	10
2.1.2.1. Choix du site	10
2.1.2.1.1. Le site	10
2.1.2.1.2. Description du site.....	11
2.1.2.1.3. Description du Lycée Neruda.....	11
2.1.2.1.4. Les valeurs de vents	12
2.1.2.2. Description de l'éolienne et de son pouvoir énergétique.....	13
2.1.2.3. Finance	15
2.1.2.4. Nuisances.....	16
2.1.2.4.1. Le Bruit	16
2.1.2.4.1. Le Visuel	16
2.2. Utilitaire de calcul	17
2.2.1. Pourquoi un utilitaire calcul.....	17
2.2.1. Notre utilitaire de calcul.....	17
Conclusions et perspectives	19
Bibliographie	20
Annexes	21

INTRODUCTION

Dans un monde où les énergies fossiles se raréfient, les énergies renouvelables apparaissent comme une solution envisageable. L'énergie éolienne est l'énergie du vent et plus spécifiquement, l'énergie tirée du vent au moyen d'un dispositif aérogénérateur comme une éolienne ou un moulin à vent.

Pour ce 4ème semestre, nous avons dû réaliser un projet à l'INSA de Rouen. Notre groupe s'est vu confier la réalisation d'un utilitaire de calcul de ressources éoliennes.

Nous avons été encadrés par un professeur de l'IUT, Mr Ludovic Henriet qui a défini les objectifs de notre travail. Dès le début, nos groupes initiaux ont été intervertis, pour permettre une meilleure adaptation des étrangers. Au début, il était prévu de réaliser un utilitaire de calcul. Finalement, il nous a été confié 2 objectifs, réaliser un projet éolien, puis un utilitaire de calcul.

Comment faire coïncider les résultats de notre projet éolien, avec un utilitaire de calcul, c'est la première question que nous nous sommes posés.

Dans un premier temps, nous nous sommes donc intéressés à Windpro et à son utilisation, logiciel très connu des experts éoliens. Nous avons vite réalisé les problèmes liés à l'installation d'une éolienne. En effet, de nombreux impératifs doivent être pris en compte. Il est vrai, nous avons pu le remarquer, l'environnement joue énormément sur le rendement de l'éolienne, comme la rugosité du sol ou encore l'exposition de l'éolienne face au vent. Tout d'abord, nous avons mis en relief l'utilisation du logiciel Windpro, puis les enjeux énergétiques et financiers d'un projet éolien à Dieppe.

Dans un second temps, nous nous sommes intéressés à la réalisation de l'utilitaire de calcul. Il nous a semblé judicieux de coder le programme en java pour faciliter la compréhension de l'utilitaire et sa lisibilité.

I. METHODOLOGIE / ORGANISATION DU TRAVAIL

1.1. Description de l'organisation adoptée pour le déroulement du travail

Lors de la première séance, il nous a été proposé de diviser le projet en deux. Un projet d'installation d'éolienne sur un site donné et un utilitaire de calcul de ressources éoliennes.

Nous avons alors décidé de séparer notre groupe en deux parties pour optimiser la réalisation des deux projets. Trois d'entre nous ont réalisé le projet d'installation d'éolienne à l'aide du logiciel Windpro et les deux autres ont programmé en JAVA l'utilitaire de calcul.

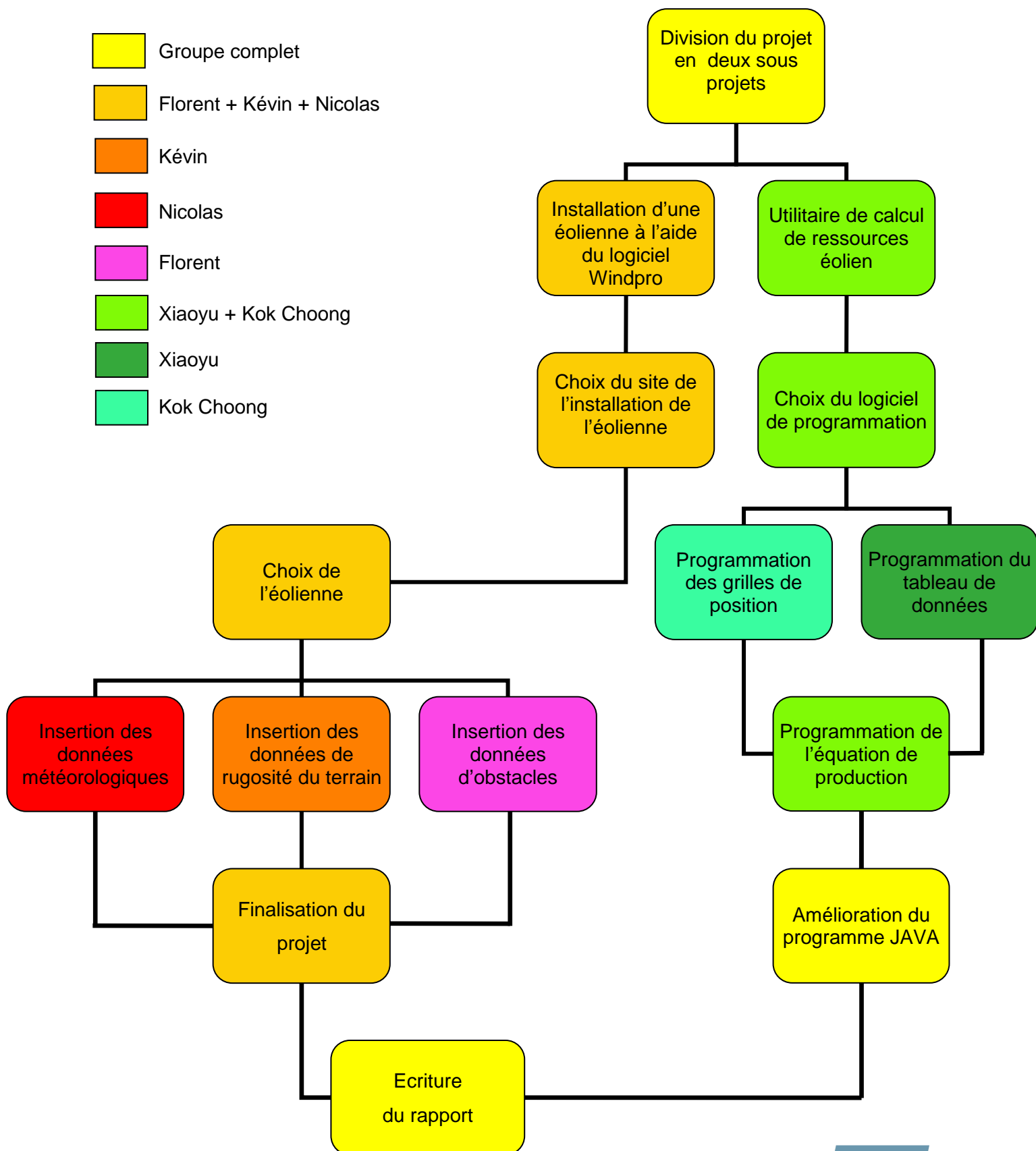
Il est évident que le choix de séparer le groupe en deux à diminuer la compréhension pour chacun des deux sujets. C'est la raison pour laquelle lors des séances avec Mr Henriet, nous mettions nos données en commun. Les sujets étant plus ou moins compliqués, nous remercions Mr Henriet pour l'aide qu'il nous a apporté.

Le sujet d'installation d'éolienne s'est effectué en trois temps. Le premier fut le choix du site. Il nous a paru intéressant de choisir le site de Dieppe pour ses caractéristiques météorologiques et environnementales. Par la suite, nous avons travaillé avec le logiciel Windpro. Puis quand les calculs effectués par le logiciel furent finis, nous avons commencé à rédiger le rapport écrit.

Pour l'utilitaire de calcul de ressources éoliennes, le travail fut plus rude. Nous avons dû programmer en JAVA des tableaux de données et des calculs de productions. Deux personnes se sont occupées de programmer et le reste du groupe à continuer le rapport écrit.

Les dernières séances de P6-3 ont été consacrées à l'amélioration du programme en JAVA et à la finalisation du rapport.

1.2. Organigramme des tâches réalisées et des étudiants concernés



II. TRAVAIL REALISE ET RESULTATS

2.1. Windpro : Projet d'éolienne à Dieppe

2.1.1. Utiliser Windpro

Windpro permet de réaliser des photomontages de parcs éoliens. Il permet aussi de calculer les émergences de bruit, la durée des ombres portées et la visibilité des éoliennes d'un parc.

Le logiciel Windpro est bien connu des experts éoliens. Il est utilisé, depuis plus de vingt ans, dans le monde entier pour l'étude des mesures de vent et pour les calculs des productibles.

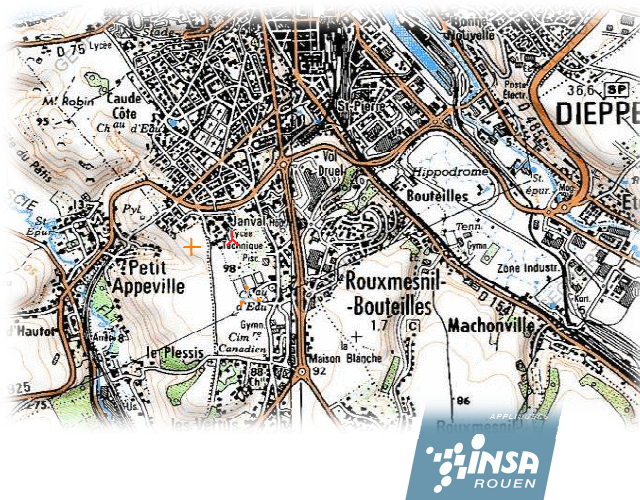
Mais Windpro est, probablement aussi, le logiciel le plus efficace et le plus facile d'utilisation pour réaliser des photomontages d'un futur parc éolien. En effet, une photo et les données de sa prise de vue suffisent pour réaliser un photomontage ultra réaliste.

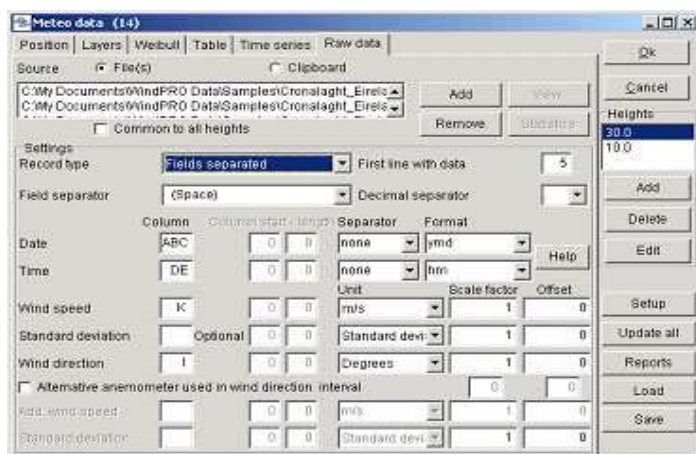
Tout d'abord avec Windpro la première chose que nous ayons eu à faire c'est d'apprendre à s'en servir. Il est vrai que Windpro offre des calculs d'une grande précision et, il est bon de le rappeler, est utilisé par la plupart des experts éoliens.



Après avoir choisi le lieu où implanter notre éolienne et en avoir fait une photo, nous avons utilisé Windpro pour calculer la productivité que fournirait notre éolienne.

Pour faire cela, nous avons entré la carte sur le logiciel, afin de pouvoir y implanter notre éolienne à l'endroit voulu. Windpro permet d'implanter une ou plusieurs éoliennes et également des objets météo qui permettent d'entrer la vitesse moyenne du vent et ainsi de calculer des énergies.

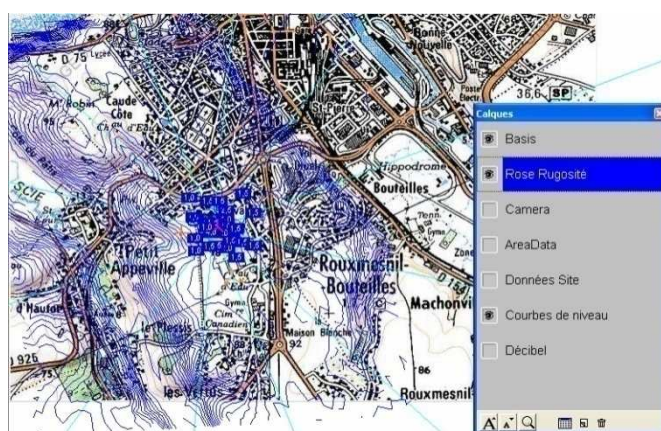




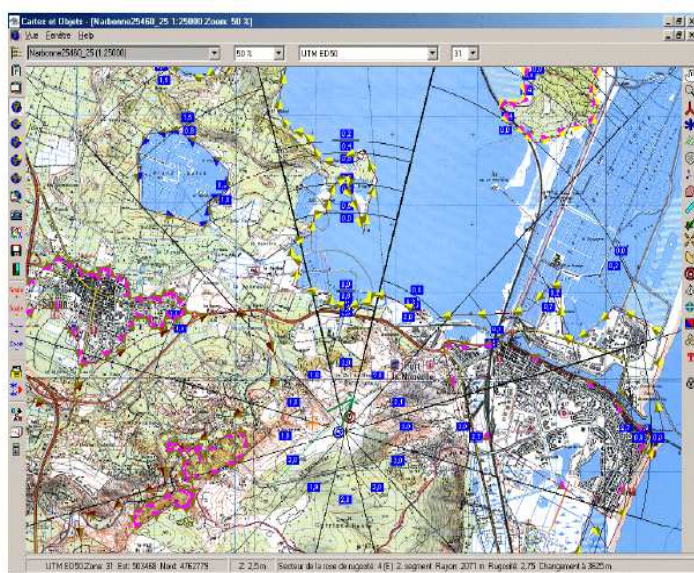
Statistique du vent (à l'aide de l'objet météo)

Le module peut travailler avec une ou plusieurs statistiques du vent qui décrivent en règle générale les conditions régionales du vent pendant une longue période. D'autres statistiques peuvent être acquises et intégrées facultativement

Heureusement pour nous, nous avons à notre disposition les données météorologiques de Dieppe, mais pour calculer le rendement de notre éolienne, il fallait également tenir compte de la qualité du terrain sur laquelle l'éolienne est implantée. Windpro permet également de prendre ce facteur en compte.



Description du terrain (rugosité)



Le terrain peut être décrit graphiquement sur un fond de carte ou bien par une carte numérique :

- Classification des rugosités du terrain et changement (rose de rugosité) dans un rayon d'environ 20km et 8 ou 12 secteurs (respectivement dépendants des statistiques du vent utilisées) ou comme import d'une carte de rugosité numérique
- Carte numérique de rugosité et/ou de hauteur, qui peuvent être numérisées directement sur le fond de carte avec différentes échelles et différents rayons.

Obstacles :

Les obstacles locaux sont définis comme rectangle directement sur la carte et sont complétés par les informations sur la hauteur et porosité

Une fois tout cela rentré dans le logiciel, il nous restait à calculer la productivité finale. De plus Windpro est un logiciel très efficace pour réaliser des photomontages de parc éolien. En effet, il nous a suffi de quelques données de la prise de vue de la photo pour réaliser un photomontage ultra réaliste.



2.1.2. Projet Dieppe

Pour mettre en pratique nos connaissances sur le logiciel Windpro, il nous a été proposé de travailler sur un projet d'installation d'éolienne à Dieppe.

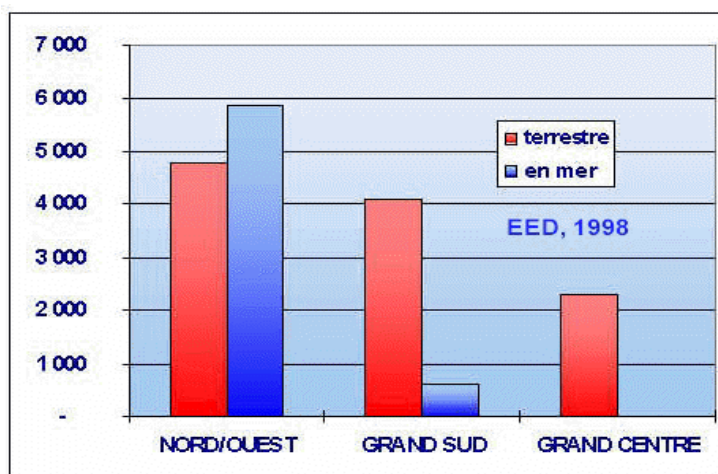
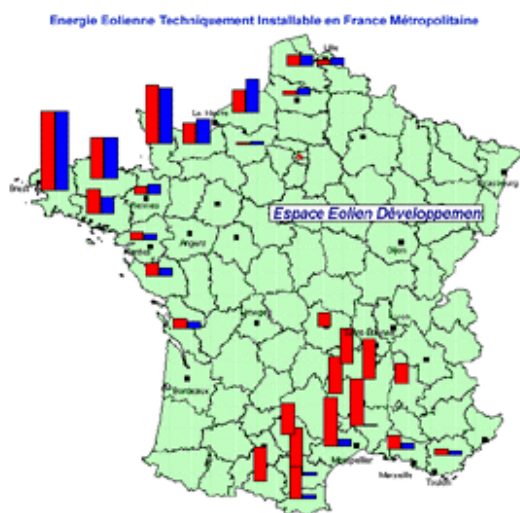
2.1.2.1. Choix du site

2.1.2.1.1 Le site

Le choix d'un site éolien se fait à partir d'une étude préalable que nous ne pouvions faire par manque de temps c'est pourquoi le choix du lycée Pablo Neruda de Dieppe nous a été donné. Malgré tout, nous allons expliquer les raisons du choix de ce site.

En effet, la région Haute Normandie possède l'un des plus fort potentiel éolien français, notamment grâce au littoral de la Seine Maritime et aux plateaux céréaliers du département de l'Eure. De plus, il s'agit de notre région.

Vous pouvez d'ailleurs y découvrir ci-dessous le potentiel de chaque région et y observer le fort potentiel du Nord-ouest en général.



Puissance potentielle en MW

A noter que ce potentiel techniquement instable ne représente que 10% de notre production électrique annuelle

Ensuite, Dieppe est une ville propice à l'implantation d'éolienne puisque cette ville possède des conditions idéales. D'ailleurs 49 projets éoliens sont en cours ce qui justifie ce choix. Enfin, le projet du lycée Pablo Neruda est un véritable projet qui se fera pour 2010.

C'est donc avec cette proximité et cette implication dûe au fait qu'il s'agisse de quelque chose de réel que nous avons commencé notre travail.

2.1.2.1.2. Description du site

Nous allons maintenant décrire le site dans de plus amples détails avec notamment ses contraintes et ses avantages.

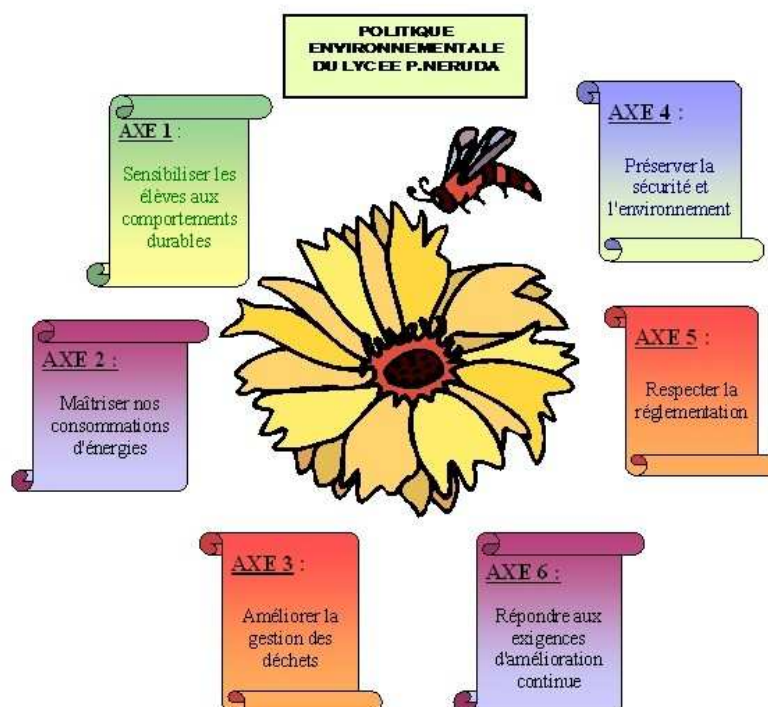
L'implantation d'un parc éolien sur un site va modifier son environnement. Elle aura un impact sur la valeur patrimoniale du site, qu'il soit artistique, original ou plus ordinaire. Il est donc important d'identifier les zones d'implantation du patrimoine culturel, afin de définir si l'implantation d'éoliennes est compatible avec la préservation de ce patrimoine.

Il en est de même pour sa faune et sa flore.

2.1.2.1.3. Description du Lycée Neruda

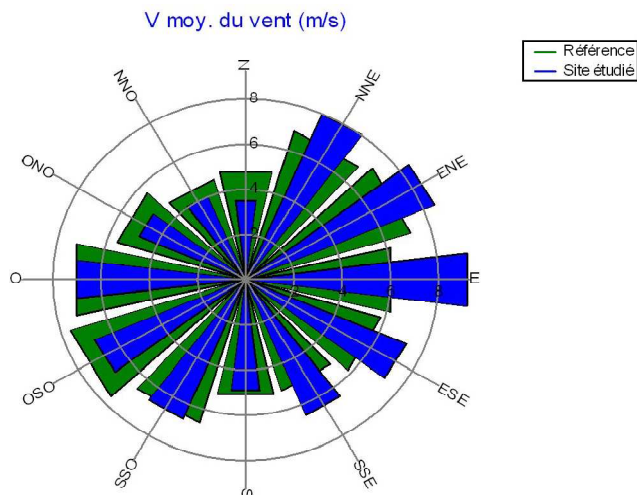
Le Lycée Pablo Neruda est le premier lycée de France et d'Europe à être certifié ISO 14001 depuis 2000. Les normes ISO 14 000 relatives aux systèmes de management environnemental aident les organisations de toutes sortes à améliorer leurs performances environnementales tout en exerçant un impact positif sur leurs résultats. Cette norme ISO 14001 est la plus courante des normes 14000 et concernent le management environnemental. Le lycée s'engage donc à répondre à plusieurs exigences en termes d'environnement, notamment celle de mettre en œuvre des actions pour satisfaire la politique environnementale. Cette norme est donc en accord et justifie notre projet éolien.

Le lycée a résumé ces exigences autour de 6 axes que vous pouvez découvrir ci-dessous :



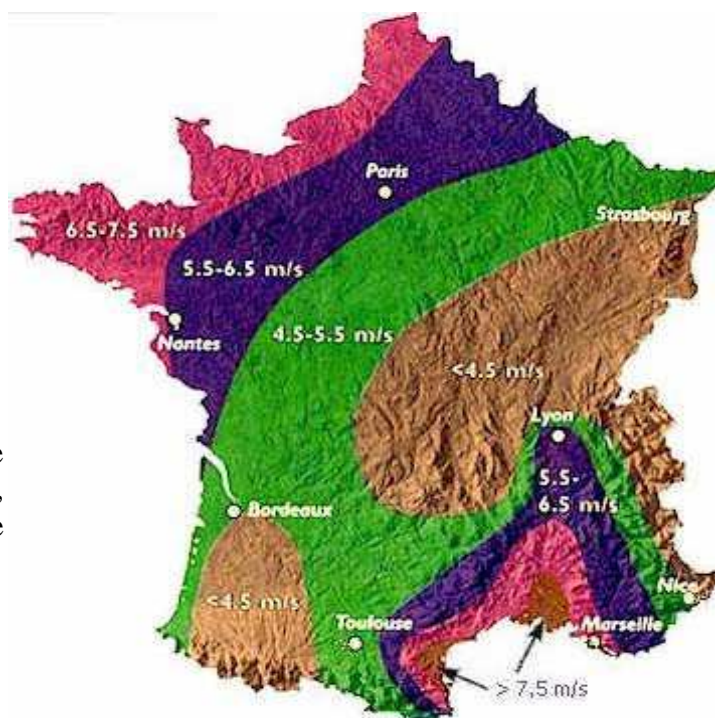
2.1.2.1.4. Les valeurs de vents

Dans la région, les vents dominants sont principalement orientés de secteur Ouest à Sud-ouest. Néanmoins, on observe aussi des vents de secteur Nord-est, assez fréquemment.

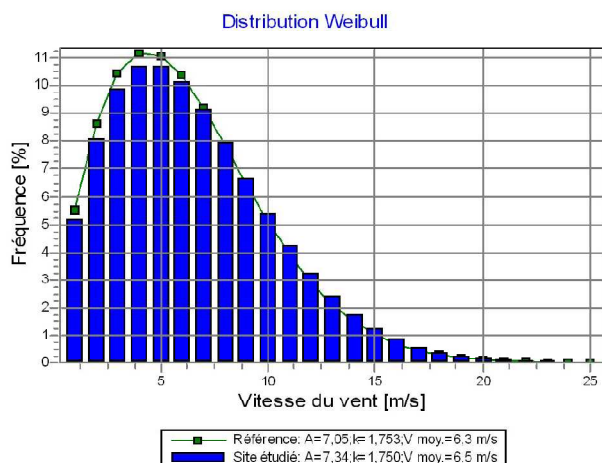


Cette rose des vents est assez représentative de la répartition des vents au niveau régional.

Comme on peut le remarquer sur cette carte de France des vitesses moyennes des vents, la région normande est une région très exposée au vent



Description de Weibull



On décrit normalement les variations du vent sur un site donné en utilisant une distribution de Weibull comme celle que vous voyez sur l'image de gauche. Sur le site en question, la vitesse moyenne du vent est de 6,50 m/s, et la forme de la courbe est déterminée par ce qu'on appelle un paramètre de forme égal à 1,750.

Ce graphe est une distribution de probabilité (distribution fréquentielle). La probabilité que le vent souffle à une vitesse quelconque (zéro inclus) étant obligatoirement égale à 100 %, la zone au dessous de la courbe sera toujours égale à exactement 1.

La distribution statistique des vitesses du vent varie d'un endroit à l'autre vu qu'elle dépend des conditions climatiques locales, le paysage et sa surface. La distribution de Weibull tend donc à varier, tant en forme qu'en valeur moyenne.

2.1.2.2. Description de l'éolienne et de son pouvoir énergétique

Fiche technique de l'éolienne :



CARACTERISTIQUES GENERALES

Type	Axe horizontal
Axe :	Horizontal
Diamètre	8
Type de mat	Treillis ou haubané
Vitesse de rotation nominale	200 tr / mn
Nombre de pales de l'hélice	3

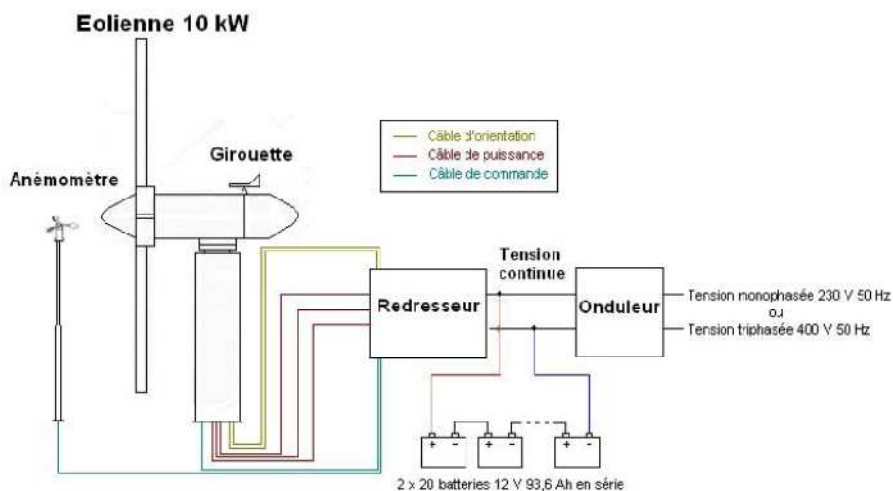
GENERATEUR

Type de génératrice	Triphasé, aimants
Tensions disponibles	110/220 V
Tension de sortie	220 V

DIMENSION

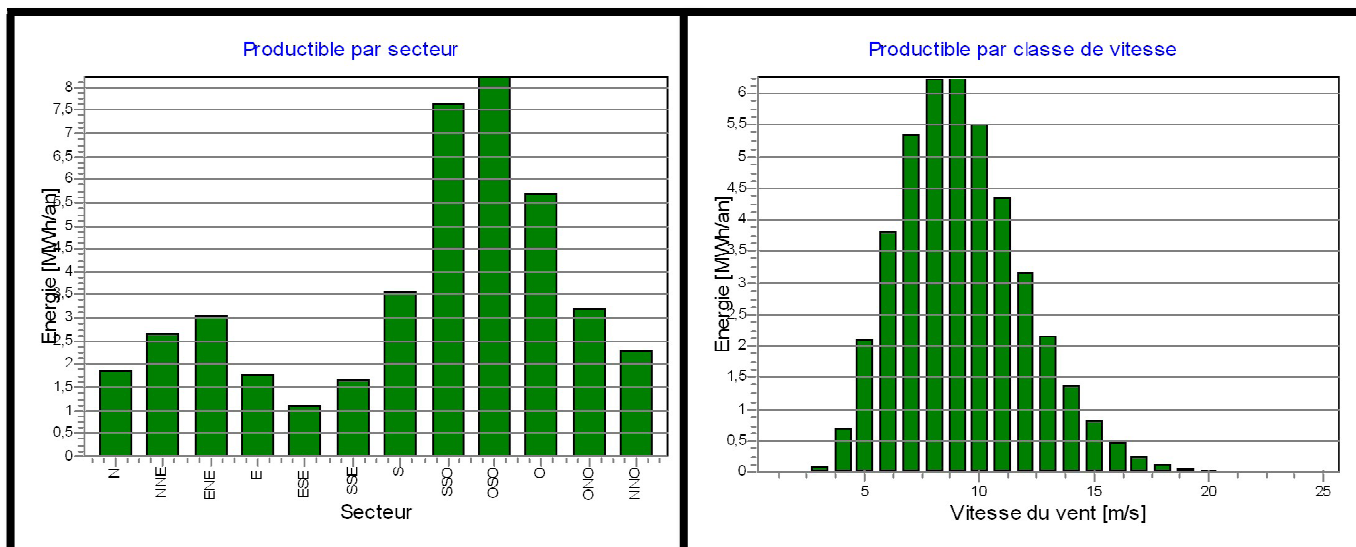
Hauteur	11,5 m
Diamètre de l'hélice	8 m
Poids total hors mât	425 kg

Schéma de connexion en autoconsommation



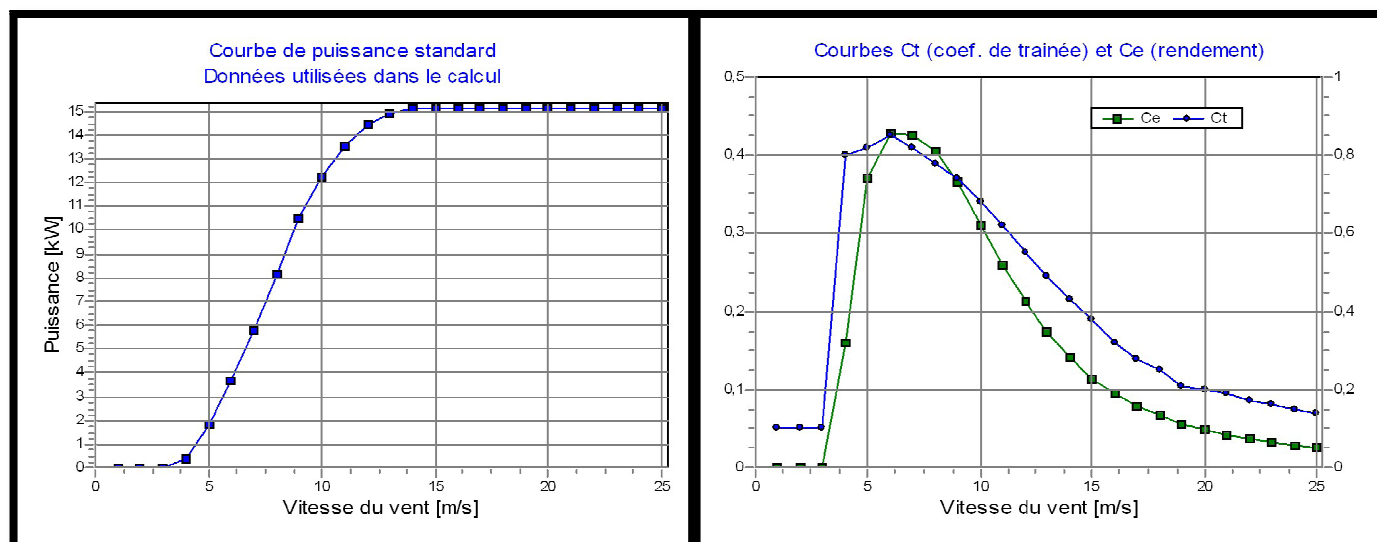
Courbes de production :

L'éolienne délivre une puissance moyenne de 26,3 KW/h par an en tenant compte de la rugosité qui est de 0,7 et d'un vent moyen de 8,4 m/s. Si les obstacles et le relief sont pris en compte la puissance moyenne est égale à 22 KW/h par an.



Les courbes de production nous montrent que l'éolienne est au maximum de production pour une vitesse de vent comprise entre 8 et 9 m/s.

Courbes de puissance :



Le graphe montre une courbe de puissance d'une éolienne de 10 kW.

On élabore les courbes de puissance à partir de mesures réalisées sur site en fixant un anémomètre sur un mât situé à proximité de l'éolienne.

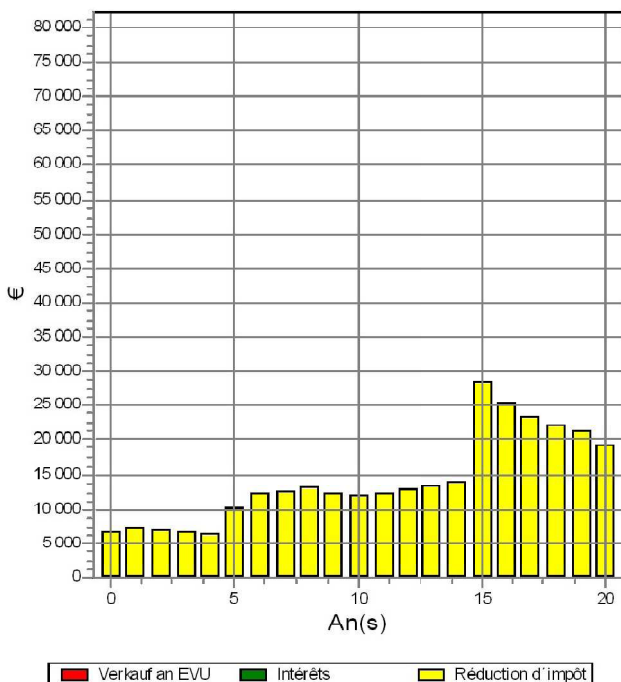
La courbe de puissance est un graphe représentant la puissance de sortie d'une éolienne à des vitesses de vent différentes.

On sait que $1 \text{ m/s} = 3.6 \text{ km/h}$. Une éolienne fournit le plus de puissance à 15 m/s soit environ 54 km/h. Plus le vent souffle plus la puissance augmente mais seulement jusqu'à 15 m/s où elle est constante. La puissance est proportionnelle à la vitesse du vent de 4m/s à 15 m/s. Si on a un vent de 10 m/s on obtient une puissance d'environ 12 kW.

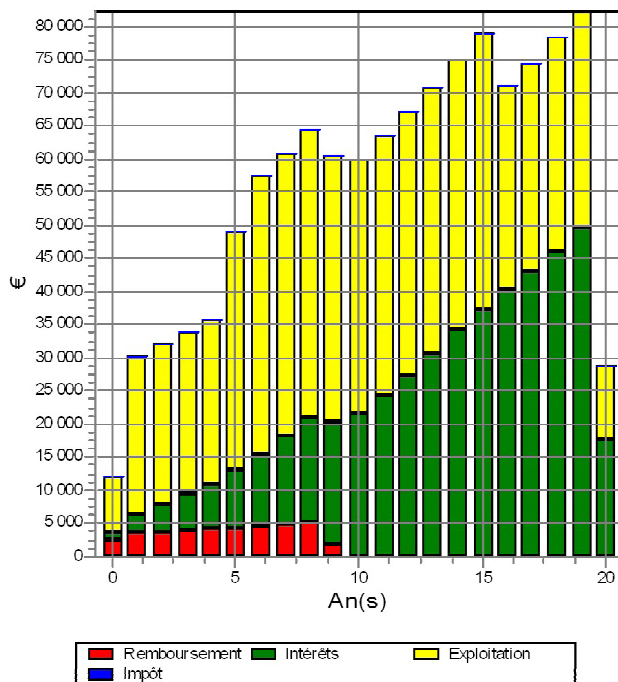
2.1.2.3. Finance

En prenant pour référence un prix du kW/h à 0.1106 € fixé par E.D.F. (Electricité De France), le lycée Neruda réalisera une économie de près de 21 000 € par an avec une production moyenne de 22 kW/h. En comparaison, une famille de 4 personnes consomme 4000 kWh par an soit près de 442 €. Le lycée Neruda pourrait donc alimenter 48 foyers de 4 personnes.

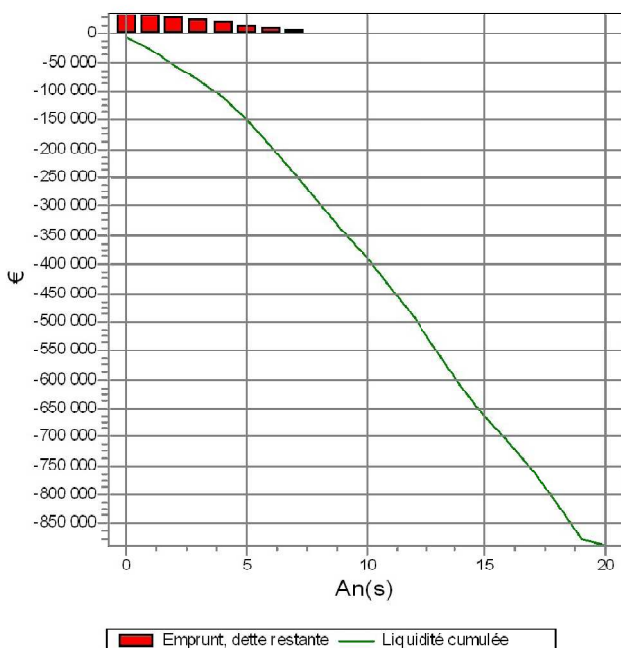
RECETTES



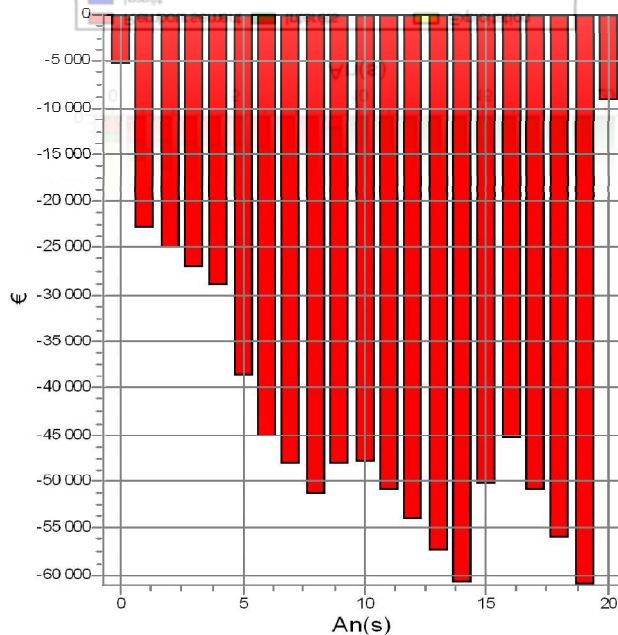
CHARGES



Dettes restant et liquidité cumulée avec impôt et financement



LIQUIDITE ANNUELLE

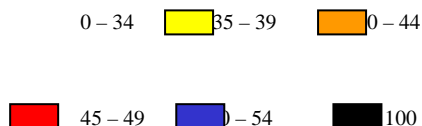


Les différentes courbes du rapport financier nous montrent que l'éolienne devient rentable au bout de 8 ans.

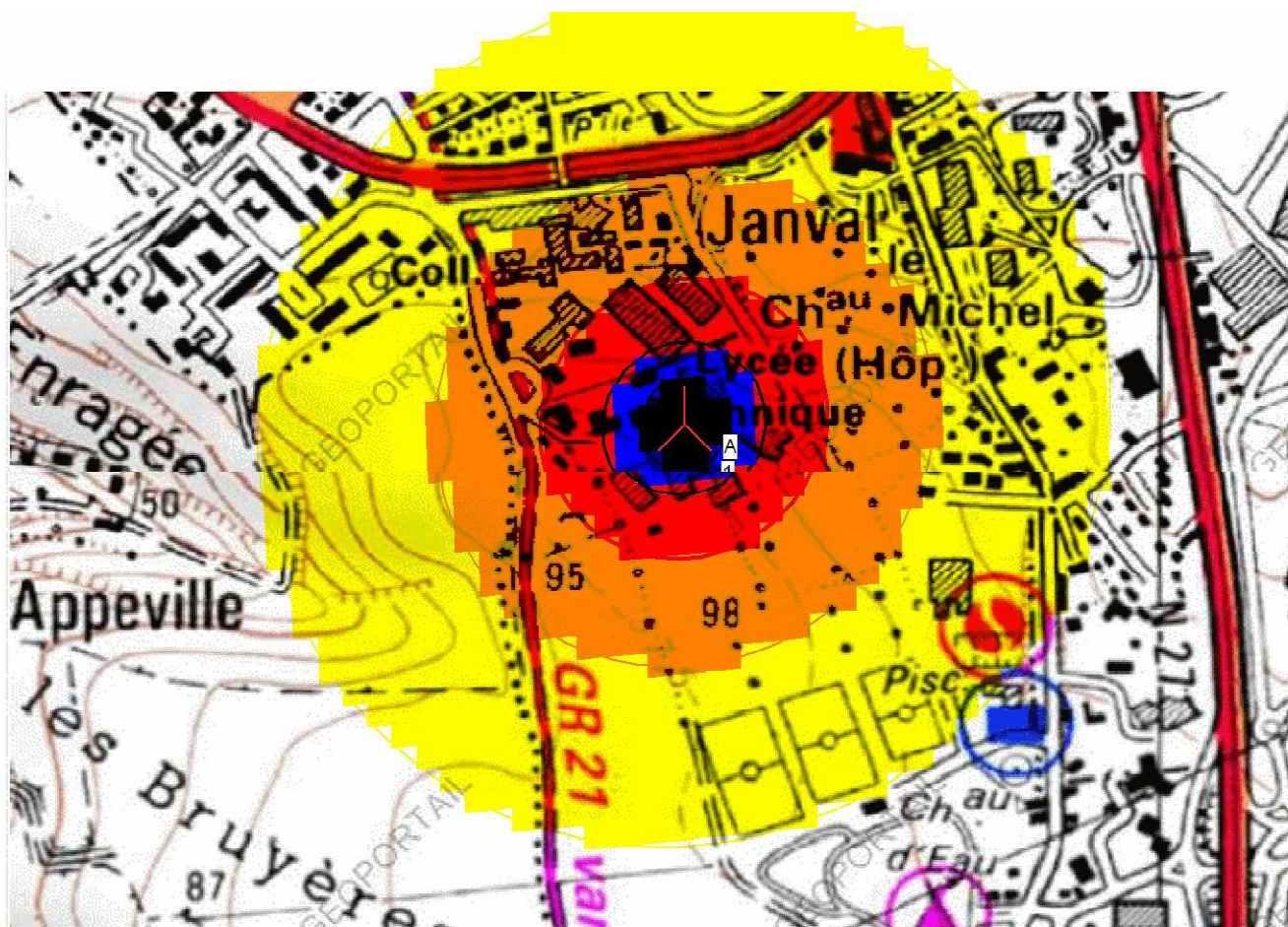
2.1.2.4. Nuisances

2.1.2.4.1. Le Bruit

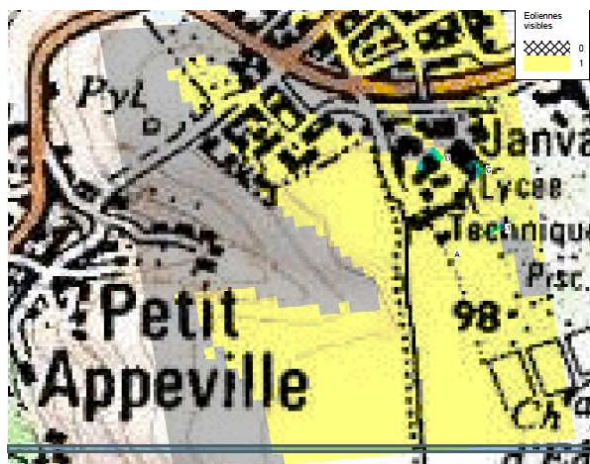
Bruit [dB(A)] :



Le bruit à proximité de l'éolienne peut être comparé au bruit que produit une route à circulation dense ou à une tronçonneuse. Dans la partie jaune du schéma ce bruit peut être comparé à celui d'une forêt.



2.1.2.4.2. Le Visuel



L'éolienne étant sur une petite butte, elle est visible à plusieurs kilomètres malgré sa plus ou moins faible hauteur. La carte IGN nous montre qu'elle ne sera visible que dans certains quartiers. L'éolienne ne devrait pas être une grande nuisance visuelle.

Windpro est un logiciel professionnel, créé pour des professionnels et utilisé par des professionnels. C'est un logiciel complet avec lequel il est assez facile de se familiariser. Nous avons en effet eu l'occasion de tester ce logiciel sur un projet concret. Windpro permet de visualiser tous les aspects de l'implantation d'une éolienne dans un secteur donnée.

2.2. Utilitaire de calcul de ressources éoliennes

2.2.1. Pourquoi un utilitaire de calcul ?

Un utilitaire de calcul est un programme informatique qui permet de calculer des prévisions de vent et la puissance produite par une éolienne ou un parc éolien. Les prévisions sont fondées sur des données de vent mesurées dans des stations de la même région.

Ce genre d'utilitaire est utilisé pour :

- Calculer l'efficacité d'un Parc éolien
- Micro-implantation d'éoliennes
- Calculs de la production d'électricité
- Cartographie des ressources de vents
- Estimation du vent
- Vent d'analyse de données

2.2.1. Notre utilitaire de calcul

Notre utilitaire est bien évidemment moins complexe que des logiciels tels que Windpro ou encore Wasp. Il a été créé dans le langage JAVA

La fonction de cet utilitaire est de calculer la distribution de Weibull et la puissance du vent généré selon les directions à partir de la vitesse du vent.

Premièrement, les différentes cases de l'utilitaire sont vides.

Il faut entrer les vitesses de vent dans les cases de la première colonne. Puis on peut cliquer sur le bouton « Calculer ».

Cette utilitaire est équipée de formules mathématiques pour calculer les fonctions indiquées au dessus des colonnes.

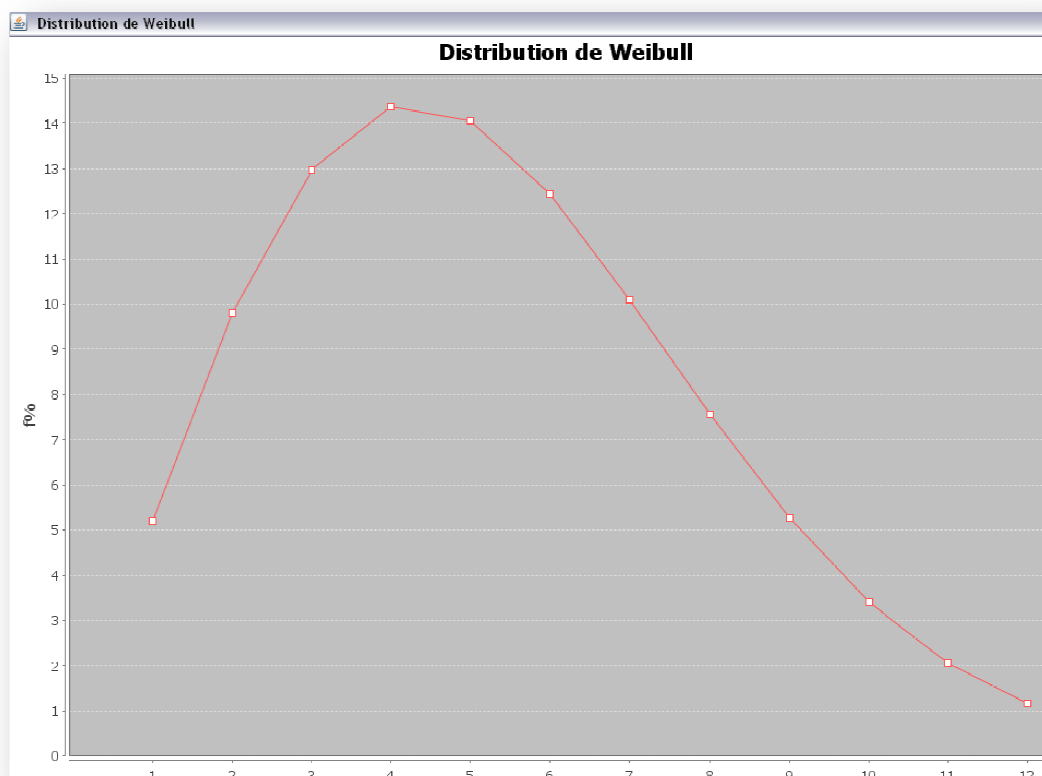
En fait la puissance du vent seule ne veut pas tout dire. Selon la loi de Betz, créée par l'Allemand Albert Betz, le maximum d'énergie cinétique récupérable est seulement de 59%. Ensuite, pour déterminer la puissance générée par une éolienne, il faut prendre en compte différents critères tels que le modèle de l'éolienne, la rugosité, le cisaillement du vent etc. Dans cette utilitaire, on a défini le coefficient de Weibull = 2.03 et le paramètre = 6 pour faciliter le calcul.

Calcul utilitaire			
Entrez la vitesse du vent	Distribution de Weibull	Puissance du vent	Puissance du vent récupérable
1	0.052049504955021086	0.6125	0.361375
2	0.09739050127409064	4.9	2.091
3	0.12670140609803505	16.5075	9.757125
4	0.14364307128855627	39.2	23.128
5	0.14055290290656711	76.5625	45.171875

Entrez la vitesse moyenne pour calculer la production d'électricité		Production d'électricité par ans (KWh)
0	0.0	La production d'électricité est calculé à partir de l'éolienne NORDEX N50 800kW
4	157680.0	
8	2233800.0	
12	5930520.0	
19	8561240.0	
25	8148520.0	

Calculer Visualiser la distribution de Weibull

De plus, cet utilitaire de calcul permet aussi de visualiser la distribution de Weibull dans une fenêtre graphique. Ce graphique est utile pour savoir la fréquence de la vitesse du vent dans un lieu choisi



CONCLUSIONS ET PERSPECTIVES

Nous avons pour objectif de réaliser un utilitaire de calculs permettant de modéliser la distribution des vents en fonction des critères du terrain. A ce travail une réalisation de projet éolien y a été ajoutée pour nous mettre en condition quasi réelle. Nous avons donc pris en compte tous les différents domaines et paramètres que des professionnels étudieraient.

En effet, notre projet d'implantation éolien nous a été inspiré d'un véritable projet qui se fera fin 2009 sur le site de Dieppe au lycée Pablo Neruda. Grâce à l'utilisation du logiciel Windpro, nous avons été capables de rentrer toutes nos données trouvées telles que la cartographie ou les vents et ainsi de mettre en œuvre un projet à caractère professionnel avec notamment le calcul du prix de revient de l'éolienne, son rendement et sa rentabilité à moyen et long terme.

Une fois cette étude finalisée, nous nous sommes attelés à créer une application JAVA permettant d'étayer nos résultats. En effet, l'utilitaire est capable de calculer et de fournir la puissance générée par notre éolienne en fonction des points *cardinaux* (N S E W) modélisée à l'aide d'une courbe.

Ce travail et d'un point de vue plus général, nous a permis de travailler en équipe comme un bureau d'étude aurait pu le faire. Cette idée est renforcée par notre projet lui-même qui nous vous le rappelons deviendra concret et sera mis en œuvre fin 2009. Sur le plan humain, il est clair que cette UV de par son fonctionnement a permis de faire connaissance et d'être réunis autour d'un objectif commun, il s'agit donc d'une bonne et première expérience pour un futur ingénieur.

Pour finir, ce projet pourra être repris et complété dans le futur avec par exemple une animation en condition réelle où on pourrait y constater les efforts subits par l'éolienne. Il serait aussi intéressant pour un cours de physique appliquée de construire une maquette éolienne et d'étudier son rendement réel et théorique.

BIBLIOGRAPHIE

La plupart des informations du dossier ont été fournies par Mr Henriet sous forme informatique.

Site :

<http://www.france-eoliennes.com/> (valide à la date du 06/06/2009)

<http://www.emd.dk/WindPRO/Introduction/> (valide à la date du 06/06/2009)

<http://en.wikipedia.org/wiki/WindPRO> (valide à la date du 06/06/2009)

http://www.actu-environnement.com/ae/fournisseur/entreprise/energie/informatique_logiciels.php4
(Valide à la date du 06/06/2009)

http://www.emd.dk/files/Formation_Paris_30-31oct_6-7-8nov07.pdf (valide à la date du 06/06/2009)

<http://www.windpower.org/fr/tour/wres/weibull.htm> (valide à la date du 06/06/2009)

<http://www.thewindpower.net/article-45-l-energie-eolienne-en-france-et-en-europe.php> (valide à la date du 06/06/2009)

<http://www.dieppe.fr/boussole/boussole.php?pg=/boussole/culture.html> (valide à la date du 06/06/2009)

<http://lycees.ac-rouen.fr/neruda/dieppe/> (valide à la date du 06/06/2009)

<http://www.region-haute-normandie.com/actions/environnement/PRE/fichiers/guide.pdf> (valide à la date du 06/06/2009)

ANNEXES

Codage du programme en JAVA sur Eclipse :

Classe intro

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class intro extends JFrame {
    private JButton start = new JButton ("Cliquer pour commencer le calcul");
    private JLabel name = new JLabel ("-----Utilitaire de calcul-----
-----");
    private JLabel left = new JLabel ();
    private JLabel right = new JLabel ();
    private JLabel left1 = new JLabel ();
    private JLabel right1 = new JLabel ();
    private JPanel panel = new JPanel ();
    private GridBagLayout grille = new GridBagLayout();
    private GridBagConstraints contraintes = new GridBagConstraints();

    private void placerDansGrille(GridBagConstraints gbc, int gx, int gy,
        int gw, int gh, int wx, int wy, int f, int a)
    {gbc.gridx = gx;    // position en x de l'objet
    gbc.gridy = gy;    // position en y de l'objet
    gbc.gridwidth = gw; // nombre de colonnes de l'objet
    gbc.gridheight = gh; // nombre de lignes de l'objet
    gbc.weightx = wx; // largeur en proportion des colonnes
    gbc.weighty = wy; // profondeur en proportion des lignes
    gbc.fill = f;
    gbc.anchor = a;}

    public intro ()
```

```

{
    setTitle("Utilitaire de calcul");
    setBounds (0,0,1050,1000);
    setDefaultCloseOperation(EXIT_ON_CLOSE);
    ImageIcon img = new ImageIcon("images/eolienne.JPG");
    JLabel image = new JLabel(img) ;

    Font police = new Font("SansSerif", Font.BOLD, 40 );
    name.setFont(police);
    name.setForeground(Color.BLUE);

    panel.setLayout(grille);
    placerDansGrille(contraintes, 1, 2, 1, 1, 40, 20,
        GridBagConstraints.BOTH, GridBagConstraints.CENTER);
    grille.setConstraints(start, contraintes);
    panel.add(start);

    placerDansGrille(contraintes, 0, 0, 3, 1, 0, 20,
        GridBagConstraints.BOTH, GridBagConstraints.CENTER);
    grille.setConstraints(name, contraintes);
    panel.add(name);

    /*placerDansGrille(contraintes, 0, 0, 1, 1, 30, 0,
        GridBagConstraints.BOTH, GridBagConstraints.CENTER);
    grille.setConstraints(left1, contraintes);
    panel.add(left1);

    placerDansGrille(contraintes, 0, 2, 1, 1, 30, 0,
        GridBagConstraints.BOTH, GridBagConstraints.CENTER);
    grille.setConstraints(right1, contraintes);
    panel.add(right1);*/

    placerDansGrille(contraintes, 0, 2, 1, 1, 30, 0,
        GridBagConstraints.BOTH, GridBagConstraints.CENTER);

```

```

grille.setConstraints(left, contraintes);
panel.add(left);
left.setBackground(Color.white);

placerDansGrille(contraintes, 2, 2, 1, 1, 30, 0,
    GridBagConstraints.BOTH, GridBagConstraints.CENTER);
grille.setConstraints(right, contraintes);
panel.add(right);
right.setBackground(Color.white);

placerDansGrille(contraintes, 0, 1, 3, 1, 0, 60,
    GridBagConstraints.BOTH, GridBagConstraints.CENTER);
grille.setConstraints(image, contraintes);
panel.add(image);
//start.setBackground(Color.white);
panel.setBackground(Color.white);

start.addActionListener(new ActionListener()
{public void actionPerformed(ActionEvent evt)
    { setVisible(false);
calcul3 h= new calcul3 ();
        });

setResizable(false);
setContentPane(panel);

setVisible(true);
}
public static void main (String arg[])
    { intro g= new intro ();
    }
}

```

Class calcul3

```
import org.jfree.chart.ChartFactory;
import org.jfree.chart.JFreeChart;
import org.jfree.chart.plot.PlotOrientation;
import org.jfree.data.category.*;
import java.awt.*;
import java.awt.event.*;

import javax.swing.*;
import java.math.*;
import org.jfree.chart.ChartFactory;
import org.jfree.chart.ChartPanel;
import org.jfree.chart.JFreeChart;
import org.jfree.chart.axis.NumberAxis;
import org.jfree.chart.plot.CategoryPlot;
import org.jfree.chart.plot.PlotOrientation;
import org.jfree.chart.renderer.category.LineAndShapeRenderer;
import org.jfree.chart.title.TextTitle;
import org.jfree.data.category.CategoryDataset;
import org.jfree.data.category.DefaultCategoryDataset;
import org.jfree.ui.ApplicationFrame;
import org.jfree.ui.HorizontalAlignment;
import org.jfree.ui.RectangleEdge;
import org.jfree.ui.RefineryUtilities;

public class calcul3 extends JFrame{
    private JTextField text1 = new JTextField (10);
    private JTextField text2 = new JTextField (10);
    private JTextField text3 = new JTextField (10);
    private JTextField text4 = new JTextField (10);
    private JTextField text5 = new JTextField (10);
```



```
private JLabel text6 = new JLabel ("Entrez la vitesse moyenne pour calculer la
production d'electricité");
private JTextField text7 = new JTextField (10);
private JTextField text8 = new JTextField (10);
private JTextField text9 = new JTextField (10);
private JTextField text10 = new JTextField (10);
private JTextField text11 = new JTextField (10);
private JTextField text12 = new JTextField (10);
private JTextField text13 = new JTextField (10);
private JTextField text14 = new JTextField (10);
private JTextField text15 = new JTextField (10);
private JTextField text16 = new JTextField (10);
private JTextField text17 = new JTextField (10);
private JLabel text18 = new JLabel (" Production d'electricité par ans (kWh)");
private JTextField text19= new JTextField (10);
private JTextField text20= new JTextField (10);
private JTextField text21 = new JTextField (10);
private JTextField text22= new JTextField (10);
private JTextField text23 = new JTextField (10);
private JTextField text24 = new JTextField (10);
private JTextField text25 = new JTextField (10);
private JTextField text26 = new JTextField (10);
private JTextField text27= new JTextField (10);
private JTextField text28= new JTextField (10);
private JTextField text29 = new JTextField (10);
private JLabel text30= new JLabel (" ");
private JLabel text31 = new JLabel ("La production d'electricité est calculé");
private JLabel text32 = new JLabel ("a partir de l'eolienne NORDEX N50 800kW");
private JLabel text33 = new JLabel ("");
private JLabel text34 = new JLabel ("");
private JLabel text35 = new JLabel ("");
private JLabel text36 = new JLabel ("");
private JTextField text37 = new JTextField (10);
private JTextField text38 = new JTextField (10);
```

```

private JTextField text39= new JTextField (10);
private JTextField text40= new JTextField (10);
private JTextField text41 = new JTextField (10);
private JLabel text42= new JLabel ("");
private JLabel text43 = new JLabel ("");
private JLabel text44 = new JLabel ("");
private JLabel text45 = new JLabel ("");
private JLabel text46 = new JLabel ("");
private JLabel text47 = new JLabel ("");
private JLabel text48 = new JLabel ("");

private JLabel label1 = new JLabel ("Entrez la vitesse du vent ....
");
private JLabel label2 = new JLabel ("                               Distribution de
Weibull");
private JLabel label3 = new JLabel ("                               Puissance du vent");
private JLabel label4 = new JLabel ("Puissance du vent recuperable");
private JLabel direction1 = new JLabel ("");
private JLabel direction2 = new JLabel ("");
private JLabel direction3 = new JLabel ("");
private JLabel direction4 = new JLabel ("");
private JLabel direction5 = new JLabel ("");
private JLabel direction6 = new JLabel ("");
private JLabel direction7 = new JLabel ("");
private JLabel direction8 = new JLabel ("");
private JLabel direction9 = new JLabel ("");
private JLabel direction10 = new JLabel ("");
private JLabel direction11 = new JLabel ("");
private JLabel direction12 = new JLabel ("");
private JLabel puissancemax = new JLabel ("");

private JButton enter = new JButton ("Calculer");
private JButton visualiser = new JButton ("Visualiser la distribution de Weibull");
private JLabel visualiser2 = new JLabel ("");

```

```

private JButton enterright = new JButton ("");
private JButton enterleft = new JButton ("");

private double coeffWeibull;
private double parametre;

private JPanel panel1 = new JPanel ();
private JPanel panel2 = new JPanel ();
private JPanel panel3 = new JPanel ();

private JPanel panel = new JPanel ();

private GridBagLayout grille = new GridBagLayout();
private GridBagConstraints contraintes = new GridBagConstraints();

private GridBagLayout grille1 = new GridBagLayout();
private GridBagConstraints contraintes1 = new GridBagConstraints();

private GridBagLayout grille2 = new GridBagLayout();
private GridBagConstraints contraintes2 = new GridBagConstraints();

private GridBagLayout grille3 = new GridBagLayout();
private GridBagConstraints contraintes3 = new GridBagConstraints();

private void placerDansGrille(GridBagConstraints gbc, int gx, int gy,
    int gw, int gh, int wx, int wy, int f, int a)
{ gbc.gridx = gx;    // position en x de l'objet
  gbc.gridy = gy;    // position en y de l'objet
  gbc.gridwidth = gw; // nombre de colonnes de l'objet
  gbc.gridheight = gh; // nombre de lignes de l'objet
  gbc.weightx = wx; // largeur en proportion des colonnes
  gbc.weighty = wy; // profondeur en proportion des lignes
  gbc.fill = f;
  gbc.anchor = a;}

```

```
public calcul3 ()
{ setTitle ("Calcul utilitaire");
  setBounds (0,0,1050,1000);
  setDefaultCloseOperation(EXIT_ON_CLOSE);
  setVisible(true);

  coeffWeibull= 2.03;
  parametre=6;

  panel.setLayout(grille);
  panel1.setLayout(grille1);
  panel2.setLayout(grille2);
  panel3.setLayout(grille3);

  label1.setBackground(Color.WHITE);
  placerDansGrille(contraintes1, 0, 0, 1, 1, 20, 100,
  GridBagConstraints.BOTH, GridBagConstraints.CENTER);
  grille1.setConstraints(label1, contraintes1);
  panel1.add(label1);

  placerDansGrille(contraintes1, 1, 0, 1, 1, 20, 0,
  GridBagConstraints.BOTH, GridBagConstraints.CENTER);
  grille1.setConstraints(label2, contraintes1);
  panel1.add(label2);

  placerDansGrille(contraintes1, 2, 0, 1, 1, 20, 0,
  GridBagConstraints.HORIZONTAL, GridBagConstraints.CENTER);
  grille1.setConstraints(label3, contraintes1);
  panel1.add(label3);

  placerDansGrille(contraintes1, 4, 0, 1, 1, 20, 0,
  GridBagConstraints.HORIZONTAL, GridBagConstraints.CENTER);
  grille1.setConstraints(label4, contraintes1);
```

```
panel1.add(label4);
```

```
placerDansGrille(contraintes1, 3, 0, 1, 1, 20, 0,  
GridBagConstraints.HORIZONTAL, GridBagConstraints.CENTER);  
grille1.setConstraints(puissancemax, contraintes1);  
panel1.add(puissancemax);
```

```
placerDansGrille(contraintes2, 0, 0, 1, 1, 20, 8,  
GridBagConstraints.BOTH, GridBagConstraints.CENTER);  
grille2.setConstraints(text1, contraintes2);  
panel2.add(text1);
```

```
placerDansGrille(contraintes2, 0, 1, 1, 1, 0, 8,  
GridBagConstraints.BOTH, GridBagConstraints.CENTER);  
grille2.setConstraints(text2, contraintes2);  
panel2.add(text2);
```

```
placerDansGrille(contraintes2, 0, 2, 1, 1, 0, 8,  
GridBagConstraints.BOTH, GridBagConstraints.CENTER);  
grille2.setConstraints(text3, contraintes2);  
panel2.add(text3);
```

```
placerDansGrille(contraintes2, 0, 3, 1, 1, 0, 8,  
GridBagConstraints.BOTH, GridBagConstraints.CENTER);  
grille2.setConstraints(text4, contraintes2);  
panel2.add(text4);
```

```
placerDansGrille(contraintes2, 0, 4, 1, 1, 0, 8,  
GridBagConstraints.BOTH, GridBagConstraints.CENTER);  
grille2.setConstraints(text5, contraintes2);  
panel2.add(text5);
```

```
placerDansGrille(contraintes2, 0, 5, 1, 1, 0, 8,  
GridBagConstraints.BOTH, GridBagConstraints.CENTER);  
grille2.setConstraints(text6, contraintes2);  
panel2.add(text6);
```

```
placerDansGrille(contraintes2, 0, 6, 1, 1, 0, 8,  
GridBagConstraints.BOTH, GridBagConstraints.CENTER);  
grille2.setConstraints(text7, contraintes2);  
panel2.add(text7);
```

```
placerDansGrille(contraintes2, 0, 7, 1, 1, 0, 8,  
GridBagConstraints.BOTH, GridBagConstraints.CENTER);  
grille2.setConstraints(text8, contraintes2);  
panel2.add(text8);
```

```
placerDansGrille(contraintes2, 0, 8, 1, 1, 0, 8,  
GridBagConstraints.BOTH, GridBagConstraints.CENTER);  
grille2.setConstraints(text9, contraintes2);  
panel2.add(text9);
```

```
placerDansGrille(contraintes2, 0, 9, 1, 1, 0, 8,  
GridBagConstraints.BOTH, GridBagConstraints.CENTER);  
grille2.setConstraints(text10, contraintes2);  
panel2.add(text10);
```

```
placerDansGrille(contraintes2, 0, 10, 1, 1, 0, 8,  
GridBagConstraints.BOTH, GridBagConstraints.CENTER);  
grille2.setConstraints(text11, contraintes2);  
panel2.add(text11);
```

```
placerDansGrille(contraintes2, 0, 11, 1, 1, 0, 8,  
GridBagConstraints.BOTH, GridBagConstraints.CENTER);  
grille2.setConstraints(text12, contraintes2);
```

```
panel2.add(text12);
```

```
placerDansGrille(contraintes2, 1, 0, 1, 1, 20, 0,  
GridBagConstraints.BOTH, GridBagConstraints.CENTER);  
grille2.setConstraints(text13, contraintes2);  
panel2.add(text13);
```

```
placerDansGrille(contraintes2, 1, 1, 1, 1, 0, 0,  
GridBagConstraints.BOTH, GridBagConstraints.CENTER);  
grille2.setConstraints(text14, contraintes2);  
panel2.add(text14);
```

```
placerDansGrille(contraintes2, 1, 2, 1, 1, 0, 0,  
GridBagConstraints.BOTH, GridBagConstraints.CENTER);  
grille2.setConstraints(text15, contraintes2);  
panel2.add(text15);
```

```
placerDansGrille(contraintes2, 1, 3, 1, 1, 0, 0,  
GridBagConstraints.BOTH, GridBagConstraints.CENTER);  
grille2.setConstraints(text16, contraintes2);  
panel2.add(text16);
```

```
placerDansGrille(contraintes2, 1, 4, 1, 1, 0, 0,  
GridBagConstraints.BOTH, GridBagConstraints.CENTER);  
grille2.setConstraints(text17, contraintes2);  
panel2.add(text17);
```

```
placerDansGrille(contraintes2, 1, 5, 1, 1, 0, 0,  
GridBagConstraints.BOTH, GridBagConstraints.CENTER);  
grille2.setConstraints(text18, contraintes2);  
panel2.add(text18);
```

```
placerDansGrille(contraintes2, 1, 6, 1, 1, 0, 0,  
GridBagConstraints.BOTH, GridBagConstraints.CENTER);
```

```
grille2.setConstraints(text19, contraintes2);  
panel2.add(text19);
```

```
placerDansGrille(contraintes2, 1, 7, 1, 1, 0, 0,  
GridBagConstraints.BOTH, GridBagConstraints.CENTER);  
grille2.setConstraints(text20, contraintes2);  
panel2.add(text20);
```

```
placerDansGrille(contraintes2, 1, 8, 1, 1, 0, 0,  
GridBagConstraints.BOTH, GridBagConstraints.CENTER);  
grille2.setConstraints(text21, contraintes2);  
panel2.add(text21);
```

```
placerDansGrille(contraintes2, 1, 9, 1, 1, 0, 0,  
GridBagConstraints.BOTH, GridBagConstraints.CENTER);  
grille2.setConstraints(text22, contraintes2);  
panel2.add(text22);
```

```
placerDansGrille(contraintes2, 1, 10, 1, 1, 0, 0,  
GridBagConstraints.BOTH, GridBagConstraints.CENTER);  
grille2.setConstraints(text23, contraintes2);  
panel2.add(text23);
```

```
placerDansGrille(contraintes2, 1, 11, 1, 1, 0, 0,  
GridBagConstraints.BOTH, GridBagConstraints.CENTER);  
grille2.setConstraints(text24, contraintes2);  
panel2.add(text24);
```

```
placerDansGrille(contraintes2, 2, 0, 1, 1, 20, 0,  
GridBagConstraints.BOTH, GridBagConstraints.CENTER);  
grille2.setConstraints(text25, contraintes2);  
panel2.add(text25);
```

```
placerDansGrille(contraintes2, 2, 1, 1, 1, 0, 0,
```



```
GridBagConstraints.BOTH, GridBagConstraints.CENTER);  
grille2.setConstraints(text26, contraintes2);  
panel2.add(text26);
```

```
placerDansGrille(contraintes2, 2, 2, 1, 1, 0, 0,  
GridBagConstraints.BOTH, GridBagConstraints.CENTER);  
grille2.setConstraints(text27, contraintes2);  
panel2.add(text27);
```

```
placerDansGrille(contraintes2, 2, 3, 1, 1, 0, 0,  
GridBagConstraints.BOTH, GridBagConstraints.CENTER);  
grille2.setConstraints(text28, contraintes2);  
panel2.add(text28);
```

```
placerDansGrille(contraintes2, 2, 4, 1, 1, 0, 0,  
GridBagConstraints.BOTH, GridBagConstraints.CENTER);  
grille2.setConstraints(text29, contraintes2);  
panel2.add(text29);
```

```
placerDansGrille(contraintes2, 2, 5, 1, 1, 0, 0,  
GridBagConstraints.BOTH, GridBagConstraints.CENTER);  
grille2.setConstraints(text30, contraintes2);  
panel2.add(text30);
```

```
placerDansGrille(contraintes2, 2, 6, 1, 1, 0, 0,  
GridBagConstraints.BOTH, GridBagConstraints.CENTER);  
grille2.setConstraints(text31, contraintes2);  
panel2.add(text31);
```

```
placerDansGrille(contraintes2, 2, 7, 1, 1, 0, 0,  
GridBagConstraints.BOTH, GridBagConstraints.CENTER);  
grille2.setConstraints(text32, contraintes2);  
panel2.add(text32);
```

```
placerDansGrille(contraintes2, 2, 8, 1, 1, 0, 0,  
GridBagConstraints.BOTH, GridBagConstraints.CENTER);  
grille2.setConstraints(text33, contraintes2);  
panel2.add(text33);
```

```
placerDansGrille(contraintes2, 2, 9, 1, 1, 0, 0,  
GridBagConstraints.BOTH, GridBagConstraints.CENTER);  
grille2.setConstraints(text34, contraintes2);  
panel2.add(text34);
```

```
placerDansGrille(contraintes2, 2, 10, 1, 1, 0, 0,  
GridBagConstraints.BOTH, GridBagConstraints.CENTER);  
grille2.setConstraints(text35, contraintes2);  
panel2.add(text35);
```

```
placerDansGrille(contraintes2, 2, 11, 1, 1, 0, 0,  
GridBagConstraints.BOTH, GridBagConstraints.CENTER);  
grille2.setConstraints(text36, contraintes2);  
panel2.add(text36);
```

```
placerDansGrille(contraintes2, 3, 0, 1, 1, 20, 0,  
GridBagConstraints.BOTH, GridBagConstraints.CENTER);  
grille2.setConstraints(text37, contraintes2);  
panel2.add(text37);
```

```
placerDansGrille(contraintes2, 3, 1, 1, 1, 0, 0,  
GridBagConstraints.BOTH, GridBagConstraints.CENTER);  
grille2.setConstraints(text38, contraintes2);  
panel2.add(text38);
```

```
placerDansGrille(contraintes2, 3, 2, 1, 1, 0, 0,  
GridBagConstraints.BOTH, GridBagConstraints.CENTER);
```

```
grille2.setConstraints(text39, contraintes2);  
panel2.add(text39);
```

```
placerDansGrille(contraintes2, 3, 3, 1, 1, 0, 0,  
GridBagConstraints.BOTH, GridBagConstraints.CENTER);  
grille2.setConstraints(text40, contraintes2);  
panel2.add(text40);
```

```
placerDansGrille(contraintes2, 3, 4, 1, 1, 0, 0,  
GridBagConstraints.BOTH, GridBagConstraints.CENTER);  
grille2.setConstraints(text41, contraintes2);  
panel2.add(text41);
```

```
placerDansGrille(contraintes2, 3, 5, 1, 1, 0, 0,  
GridBagConstraints.BOTH, GridBagConstraints.CENTER);  
grille2.setConstraints(text42, contraintes2);  
panel2.add(text42);
```

```
placerDansGrille(contraintes2, 3, 6, 1, 1, 0, 0,  
GridBagConstraints.BOTH, GridBagConstraints.CENTER);  
grille2.setConstraints(text43, contraintes2);  
panel2.add(text43);
```

```
placerDansGrille(contraintes2, 3, 7, 1, 1, 0, 0,  
GridBagConstraints.BOTH, GridBagConstraints.CENTER);  
grille2.setConstraints(text44, contraintes2);  
panel2.add(text44);
```

```
placerDansGrille(contraintes2, 3, 8, 1, 1, 0, 0,  
GridBagConstraints.BOTH, GridBagConstraints.CENTER);  
grille2.setConstraints(text45, contraintes2);  
panel2.add(text45);
```

```
placerDansGrille(contraintes2, 3, 9, 1, 1, 0, 0,  
GridBagConstraints.BOTH, GridBagConstraints.CENTER);  
grille2.setConstraints(text46, contraintes2);  
panel2.add(text46);
```

```
placerDansGrille(contraintes2, 3, 10, 1, 1, 0, 0,  
GridBagConstraints.BOTH, GridBagConstraints.CENTER);  
grille2.setConstraints(text47, contraintes2);  
panel2.add(text47);
```

```
placerDansGrille(contraintes2, 3, 11, 1, 1, 0, 0,  
GridBagConstraints.BOTH, GridBagConstraints.CENTER);  
grille2.setConstraints(text48, contraintes2);  
panel2.add(text48);
```

```
placerDansGrille(contraintes2, 4, 0, 1, 1, 20, 0,  
GridBagConstraints.BOTH, GridBagConstraints.CENTER);  
grille2.setConstraints(direction1, contraintes2);  
panel2.add(direction1);
```

```
placerDansGrille(contraintes2, 4, 1, 1, 1, 0, 0,  
GridBagConstraints.BOTH, GridBagConstraints.CENTER);  
grille2.setConstraints(direction2, contraintes2);  
panel2.add(direction2);
```

```
placerDansGrille(contraintes2, 4, 2, 1, 1, 0, 0,  
GridBagConstraints.BOTH, GridBagConstraints.CENTER);  
grille2.setConstraints(direction3, contraintes2);  
panel2.add(direction3);
```

```
placerDansGrille(contraintes2, 4, 3, 1, 1, 0, 0,  
GridBagConstraints.BOTH, GridBagConstraints.CENTER);
```

```
grille2.setConstraints(direction4, contraintes2);  
panel2.add(direction4);
```

```
placerDansGrille(contraintes2, 4, 4, 1, 1, 0, 0,  
GridBagConstraints.BOTH, GridBagConstraints.CENTER);  
grille2.setConstraints(direction5, contraintes2);  
panel2.add(direction5);
```

```
placerDansGrille(contraintes2, 4, 5, 1, 1, 0, 0,  
GridBagConstraints.BOTH, GridBagConstraints.CENTER);  
grille2.setConstraints(direction6, contraintes2);  
panel2.add(direction6);
```

```
placerDansGrille(contraintes2, 4, 6, 1, 1, 0, 0,  
GridBagConstraints.BOTH, GridBagConstraints.CENTER);  
grille2.setConstraints(direction7, contraintes2);  
panel2.add(direction7);
```

```
placerDansGrille(contraintes2, 4, 7, 1, 1, 0, 0,  
GridBagConstraints.BOTH, GridBagConstraints.CENTER);  
grille2.setConstraints(direction8, contraintes2);  
panel2.add(direction8);
```

```
placerDansGrille(contraintes2, 4, 8, 1, 1, 0, 0,  
GridBagConstraints.BOTH, GridBagConstraints.CENTER);  
grille2.setConstraints(direction9, contraintes2);  
panel2.add(direction9);
```

```
placerDansGrille(contraintes2, 4, 9, 1, 1, 0, 0,  
GridBagConstraints.BOTH, GridBagConstraints.CENTER);  
grille2.setConstraints(direction10, contraintes2);  
panel2.add(direction10);
```

```
placerDansGrille(contraintes2, 4, 10, 1, 1, 0, 0,
```

```
GridBagConstraints.BOTH, GridBagConstraints.CENTER);  
grille2.setConstraints(direction11, contraintes2);  
panel2.add(direction11);
```

```
placerDansGrille(contraintes2, 4, 11, 1, 1, 0, 0,  
GridBagConstraints.BOTH, GridBagConstraints.CENTER);  
grille2.setConstraints(direction12, contraintes2);  
panel2.add(direction12);
```

```
placerDansGrille(contraintes3, 1, 0, 1, 1, 20, 100,  
GridBagConstraints.BOTH, GridBagConstraints.CENTER);  
grille3.setConstraints(enter, contraintes3);  
panel3.add(enter);
```

```
placerDansGrille(contraintes3, 2, 0, 1, 1, 35, 0,  
GridBagConstraints.BOTH, GridBagConstraints.CENTER);  
grille3.setConstraints(visualiser, contraintes3);  
panel3.add(visualiser);
```

```
placerDansGrille(contraintes3, 0, 0, 1, 1, 5, 0,  
GridBagConstraints.BOTH, GridBagConstraints.CENTER);  
grille3.setConstraints(enterleft, contraintes3);  
panel3.add(enterleft);
```

```
placerDansGrille(contraintes3, 4, 0, 1, 1, 5, 0,  
GridBagConstraints.BOTH, GridBagConstraints.CENTER);  
grille3.setConstraints(enterright, contraintes3);  
panel3.add(enterright);
```

```
placerDansGrille(contraintes3, 3, 0, 1, 1, 35, 0,  
GridBagConstraints.BOTH, GridBagConstraints.CENTER);  
grille3.setConstraints(visualiser2, contraintes3);
```

```
panel3.add(visualiser2);
```

```
placerDansGrille(contraintes, 0, 0, 1, 1, 100, 15,
GridBagConstraints.BOTH, GridBagConstraints.NORTH);
grille.setConstraints(panel1, contraintes);
panel.add(panel1);
```

```
placerDansGrille(contraintes, 0, 1, 1, 1, 0, 70,
GridBagConstraints.BOTH, GridBagConstraints.CENTER);
grille.setConstraints(panel2, contraintes);
panel.add(panel2);
```

```
placerDansGrille(contraintes, 0, 2, 1, 1, 0, 15,
GridBagConstraints.BOTH, GridBagConstraints.CENTER);
grille.setConstraints(panel3, contraintes);
panel.add(panel3);
```

```
enterleft.setBackground(Color.GREEN);
enterright.setBackground(Color.GREEN);
```

```
enter.addActionListener(new ActionListener(){
private double densite=1.225;
```

```
public void actionPerformed(ActionEvent arg0) {
```

```
String t1= text1.getText();
```

```
double s= Double.parseDouble(t1);
```

```
double a= (Math.pow(s/parametre,coeffWeibull-1)*Math.exp(-
Math.pow(s/parametre,coeffWeibull))*coeffWeibull/parametre);
```

```
String str1=Double.toString(a);
```

```
text13.setText(str1);  
double p= 0.5*densite*s*s*s;  
String str2=Double.toString(p);  
text25.setText(str2);  
double p1= 0.59*p;  
String str3=Double.toString(p1);  
text37.setText(str3);
```

```
t1= text2.getText();  
s= Double.parseDouble(t1);  
a=(Math.pow(s/parametre,coeffWeibull-1)*Math.exp(-  
Math.pow(s/parametre,coeffWeibull))*coeffWeibull/parametre);  
str1=Double.toString(a);  
text14.setText(str1);  
p= 0.5*densite*s*s*s;  
str2=Double.toString(p);  
text26.setText(str2);  
p1= 0.59*p;  
str3=Double.toString(p1);  
text38.setText(str3);
```

```
t1= text3.getText();  
s= Double.parseDouble(t1);  
a=(Math.pow(s/parametre,coeffWeibull-1)*Math.exp(-  
Math.pow(s/parametre,coeffWeibull))*coeffWeibull/parametre);  
str1=Double.toString(a);  
text15.setText(str1);  
p= 0.5*densite*s*s*s;  
str2=Double.toString(p);  
text27.setText(str2);  
p1= 0.59*p;  
str3=Double.toString(p1);  
text39.setText(str3);
```



```

t1= text4.getText();
s= Double.parseDouble(t1);
a=
    (Math.pow(s/parametre,coeffWeibull-1)*Math.exp(-
    Math.pow(s/parametre,coeffWeibull))*coeffWeibull/parametre);
str1=Double.toString(a);
text16.setText(str1);
p= 0.5*densite*s*s*s;
str2=Double.toString(p);
text28.setText(str2);
p1= 0.59*p;
str3=Double.toString(p1);
text40.setText(str3);

```

```

t1= text5.getText();
s= Double.parseDouble(t1);
a=
    (Math.pow(s/parametre,coeffWeibull-1)*Math.exp(-
    Math.pow(s/parametre,coeffWeibull))*coeffWeibull/parametre);
str1=Double.toString(a);
text17.setText(str1);
p= 0.5*densite*s*s*s;
str2=Double.toString(p);
text29.setText(str2);
p1= 0.59*p;
str3=Double.toString(p1);
text41.setText(str3);

```

```

t1= text7.getText();
s= Double.parseDouble(t1);
if (s==0)
{ a=0; }
if (s==1)
{ a=0;}

```

```
if (s==2)
{ a=0; }
if (s==3)
{ a=0;}
if (s==4)
{ a=18*8760; }
if (s==5)
{ a=55*8760;}
if (s==6)
{ a=102*8760; }
if (s==7)
{ a=157*8760;}
if (s==8)
{ a=255*8760; }
if (s==9)
{ a=367*8760;}
if (s==10)
{ a=487*8760; }
if (s==11)
{ a=595*8760;}
if (s==12)
{ a=677*8760; }
if (s==13)
{ a=735*8760;}
if (s==14)
{ a=779*8760; }
if (s==15)
{ a=797*8760;}
if (s==16)
{ a=801*8760; }
if (s==17)
{ a=788*8760;}
if (s==18)
{ a=769*8760; }
```

```
if (s==19)
{ a=749*8760;}
if (s==20)
{ a=733*8760; }
if (s==21)
{ a=717*8760;}
if (s==22)
{ a=705*8760; }
if (s==23)
{ a=701*8760;}
if (s==24)
{ a=700*8760; }
if (s==25)
{ a=702*8760;}
```

```
str1=Double.toString(a);
text19.setText(str1);
```

```
t1= text8.getText();
s= Double.parseDouble(t1);
if (s==0)
{ a=0; }
if (s==1)
{ a=0;}
if (s==2)
{ a=0; }
if (s==3)
{ a=0;}
if (s==4)
{ a=18*8760; }
if (s==5)
{ a=55*8760;}
if (s==6)
```

```
{ a=102*8760; }  
if (s==7)  
{ a=157*8760;}  
if (s==8)  
{ a=255*8760; }  
if (s==9)  
{ a=367*8760;}  
if (s==10)  
{ a=487*8760; }  
if (s==11)  
{ a=595*8760;}  
if (s==12)  
{ a=677*8760; }  
if (s==13)  
{ a=735*8760;}  
if (s==14)  
{ a=779*8760; }  
if (s==15)  
{ a=797*8760;}  
if (s==16)  
{ a=801*8760; }  
if (s==17)  
{ a=788*8760;}  
if (s==18)  
{ a=769*8760; }  
if (s==19)  
{ a=749*8760;}  
if (s==20)  
{ a=733*8760; }  
if (s==21)  
{ a=717*8760;}  
if (s==22)  
{ a=705*8760; }  
if (s==23)
```

```
{ a=701*8760;}  
if (s==24)  
{ a=700*8760; }  
if (s==25)  
{ a=702*8760;}
```

```
str1=Double.toString(a);  
text20.setText(str1);
```

```
t1= text9.getText();  
s= Double.parseDouble(t1);  
if (s==0)  
{ a=0; }  
if (s==1)  
{ a=0;}  
if (s==2)  
{ a=0; }  
if (s==3)  
{ a=0;}  
if (s==4)  
{ a=18*8760; }  
if (s==5)  
{ a=55*8760;}  
if (s==6)  
{ a=102*8760; }  
if (s==7)  
{ a=157*8760;}  
if (s==8)  
{ a=255*8760; }  
if (s==9)  
{ a=367*8760;}  
if (s==10)  
{ a=487*8760; }
```

```
if (s==11)
{ a=595*8760;}
if (s==12)
{ a=677*8760; }
if (s==13)
{ a=735*8760;}
if (s==14)
{ a=779*8760; }
if (s==15)
{ a=797*8760;}
if (s==16)
{ a=801*8760; }
if (s==17)
{ a=788*8760;}
if (s==18)
{ a=769*8760; }
if (s==19)
{ a=749*8760;}
if (s==20)
{ a=733*8760; }
if (s==21)
{ a=717*8760;}
if (s==22)
{ a=705*8760; }
if (s==23)
{ a=701*8760;}
if (s==24)
{ a=700*8760; }
if (s==25)
{ a=702*8760;}
str1=Double.toString(a);
text21.setText(str1);
```

```
t1= text10.getText();
s= Double.parseDouble(t1);
if (s==0)
{ a=0; }
if (s==1)
{ a=0;}
if (s==2)
{ a=0; }
if (s==3)
{ a=0;}
if (s==4)
{ a=18*8760; }
if (s==5)
{ a=55*8760;}
if (s==6)
{ a=102*8760; }
if (s==7)
{ a=157*8760;}
if (s==8)
{ a=255*8760; }
if (s==9)
{ a=367*8760;}
if (s==10)
{ a=487*8760; }
if (s==11)
{ a=595*8760;}
if (s==12)
{ a=677*8760; }
if (s==13)
{ a=735*8760;}
if (s==14)
{ a=779*8760; }
if (s==15)
{ a=797*8760;}
```

```
if (s==16)
{ a=801*8760; }
if (s==17)
{ a=788*8760;}
if (s==18)
{ a=769*8760; }
if (s==19)
{ a=749*8760;}
if (s==20)
{ a=733*8760; }
if (s==21)
{ a=717*8760;}
if (s==22)
{ a=705*8760; }
if (s==23)
{ a=701*8760;}
if (s==24)
{ a=700*8760; }
if (s==25)
{ a=702*8760;}
str1=Double.toString(a);
text22.setText(str1);
```

```
t1= text11.getText();
s= Double.parseDouble(t1);
if (s==0)
{ a=0; }
if (s==1)
{ a=0;}
if (s==2)
{ a=0; }
if (s==3)
{ a=0;}
```



```
if (s==4)
{ a=18*8760; }
if (s==5)
{ a=55*8760;}
if (s==6)
{ a=102*8760; }
if (s==7)
{ a=157*8760;}
if (s==8)
{ a=255*8760; }
if (s==9)
{ a=367*8760;}
if (s==10)
{ a=487*8760; }
if (s==11)
{ a=595*8760;}
if (s==12)
{ a=677*8760; }
if (s==13)
{ a=735*8760;}
if (s==14)
{ a=779*8760; }
if (s==15)
{ a=797*8760;}
if (s==16)
{ a=801*8760; }
if (s==17)
{ a=788*8760;}
if (s==18)
{ a=769*8760; }
if (s==19)
{ a=749*8760;}
if (s==20)
{ a=733*8760; }
```

```
if (s==21)
{ a=717*8760;}
if (s==22)
{ a=705*8760; }
if (s==23)
{ a=701*8760;}
if (s==24)
{ a=700*8760; }
if (s==25)
{ a=702*8760;}
str1=Double.toString(a);
text23.setText(str1);
```

```
t1= text12.getText();
s= Double.parseDouble(t1);
if (s==0)
{ a=0; }
if (s==1)
{ a=0;}
if (s==2)
{ a=0; }
if (s==3)
{ a=0;}
if (s==4)
{ a=18*8760; }
if (s==5)
{ a=55*8760;}
if (s==6)
{ a=102*8760; }
if (s==7)
{ a=157*8760;}
if (s==8)
{ a=255*8760; }
```

```
if (s==9)
{ a=367*8760;}
if (s==10)
{ a=487*8760; }
if (s==11)
{ a=595*8760;}
if (s==12)
{ a=677*8760; }
if (s==13)
{ a=735*8760;}
if (s==14)
{ a=779*8760; }
if (s==15)
{ a=797*8760;}
if (s==16)
{ a=801*8760; }
if (s==17)
{ a=788*8760;}
if (s==18)
{ a=769*8760; }
if (s==19)
{ a=749*8760;}
if (s==20)
{ a=733*8760; }
if (s==21)
{ a=717*8760;}
if (s==22)
{ a=705*8760; }
if (s==23)
{ a=701*8760;}
if (s==24)
{ a=700*8760; }
if (s==25)
{ a=702*8760;}
```

```

str1=Double.toString(a);
text24.setText(str1);

});

visualiser.addActionListener(new ActionListener()
{public void actionPerformed(ActionEvent evt)
{ //setVisible(false);
distribution h= new distribution ("Distribution de Weibull");
h.pack();
RefineryUtilities.centerFrameOnScreen(h);
h.setVisible(true);
}});

/*visualiser2.addActionListener(new ActionListener()
{public void actionPerformed(ActionEvent evt)
{ setVisible(false);
distribution2 h= new distribution2 ("Puissance g n r  par l'eolienne Nordex");
h.pack();
RefineryUtilities.centerFrameOnScreen(h);
h.setVisible(true);
}});*/

panel2.setForeground(Color.GREEN);
panel1.setBackground(Color.GREEN);
setContentPane(panel);

}

public static void main (String arg[])

```

```

        { calcul3 g= new calcul3 ();

        }
}

```

Classe distribution

```

import java.awt.Color;
import java.awt.Dimension;
import java.awt.Font;
import javax.swing.JPanel;
import org.jfree.chart.ChartFactory;
import org.jfree.chart.ChartPanel;
import org.jfree.chart.JFreeChart;
import org.jfree.chart.axis.NumberAxis;
import org.jfree.chart.plot.CategoryPlot;
import org.jfree.chart.plot.PlotOrientation;
import org.jfree.chart.renderer.category.LineAndShapeRenderer;
import org.jfree.chart.title.TextTitle;
import org.jfree.data.category.CategoryDataset;
import org.jfree.data.category.DefaultCategoryDataset;
import org.jfree.ui.ApplicationFrame;
import org.jfree.ui.HorizontalAlignment;
import org.jfree.ui.RectangleEdge;
import org.jfree.ui.RefineryUtilities;

public class distribution extends ApplicationFrame {

public distribution(String title) {
super("Distribution de Weibull");
    CategoryDataset dataset = createDataset();
    JFreeChart chart = createChart(dataset);
    ChartPanel chartPanel = new ChartPanel(chart);

```

```

chartPanel.setPreferredSize(new Dimension(1000, 800));
setContentPane(chartPanel);
}

```

```

private static CategoryDataset createDataset() {

```

```

    double coeffWeibull= 2.03;

```

```

    double parametre=6;

```

```

    DefaultCategoryDataset dataset = new DefaultCategoryDataset();

```

```

    double a1 =1;

```

```

    double a2 =2;

```

```

    double a3 =3;

```

```

    double a4 =4;

```

```

    double a5 =5;

```

```

    double a6 =6;

```

```

    double a7 =7;

```

```

    double a8 =8;

```

```

    double a9 =9;

```

```

    double a10 =10;

```

```

    double a11 =11;

```

```

    double a12 =12;

```

```

    double x1= 100*(Math.pow(a1/parametre,coeffWeibull-1)*Math.exp(-
    Math.pow(a1/parametre,coeffWeibull))*coeffWeibull/parametre);

```

```

    double x2= 100*(Math.pow(a2/parametre,coeffWeibull-1)*Math.exp(-
    Math.pow(a2/parametre,coeffWeibull))*coeffWeibull/parametre);

```

```

    double x3= 100*(Math.pow(a3/parametre,coeffWeibull-1)*Math.exp(-
    Math.pow(a3/parametre,coeffWeibull))*coeffWeibull/parametre);

```

```

    double x4= 100*(Math.pow(a4/parametre,coeffWeibull-1)*Math.exp(-
    Math.pow(a4/parametre,coeffWeibull))*coeffWeibull/parametre);

```

```

    double x5= 100*(Math.pow(a5/parametre,coeffWeibull-1)*Math.exp(-
    Math.pow(a5/parametre,coeffWeibull))*coeffWeibull/parametre);

```

```

    double x6= 100*(Math.pow(a6/parametre,coeffWeibull-1)*Math.exp(-
    Math.pow(a6/parametre,coeffWeibull))*coeffWeibull/parametre);

```

```

    double x7= 100*(Math.pow(a7/parametre,coeffWeibull-1)*Math.exp(-
    Math.pow(a7/parametre,coeffWeibull))*coeffWeibull/parametre);

```

```

double          x8=          100*(Math.pow(a8/parametre,coeffWeibull-1)*Math.exp(-
Math.pow(a8/parametre,coeffWeibull))*coeffWeibull/parametre);
double          x9=          100*(Math.pow(a9/parametre,coeffWeibull-1)*Math.exp(-
Math.pow(a9/parametre,coeffWeibull))*coeffWeibull/parametre);
double          x10=         100*(Math.pow(a10/parametre,coeffWeibull-1)*Math.exp(-
Math.pow(a10/parametre,coeffWeibull))*coeffWeibull/parametre);
double          x11=         100*(Math.pow(a11/parametre,coeffWeibull-1)*Math.exp(-
Math.pow(a11/parametre,coeffWeibull))*coeffWeibull/parametre);
double          x12=         100*(Math.pow(a12/parametre,coeffWeibull-1)*Math.exp(-
Math.pow(a12/parametre,coeffWeibull))*coeffWeibull/parametre);

```

```

dataset.addValue(x1, "f%", "1");
dataset.addValue(x2, "f%", "2");
dataset.addValue(x3, "f%", "3");
dataset.addValue(x4, "f%", "4");
dataset.addValue(x5, "f%", "5");
dataset.addValue(x6, "f%", "6");
dataset.addValue(x7, "f%", "7");
dataset.addValue(x8, "f%", "8");
dataset.addValue(x9, "f%", "9");
dataset.addValue(x10, "f%", "10");
dataset.addValue(x11, "f%", "11");
dataset.addValue(x12, "f%", "12");
return dataset;
}

```

```

private static JFreeChart createChart(CategoryDataset dataset) {

```

```

JFreeChart chart = ChartFactory.createLineChart(
"Distribution de Weibull",
"Vitesse du vent (m/s)", // domain axis label
"f%", // range axis label
dataset, // data
PlotOrientation.VERTICAL, // orientation

```

```

false, // include legend
true, // tooltips
false // urls
);

chart.setBackgroundPaint(Color.white);
CategoryPlot plot = (CategoryPlot) chart.getPlot();
plot.setBackgroundPaint(Color.lightGray);
plot.setRangeGridlinePaint(Color.white);
// customise the range axis...
NumberAxis rangeAxis = (NumberAxis) plot.getRangeAxis();
rangeAxis.setStandardTickUnits(NumberAxis.createIntegerTickUnits());
// customise the renderer...
LineAndShapeRenderer renderer
= (LineAndShapeRenderer) plot.getRenderer();
renderer.setShapesVisible(true);
renderer.setDrawOutlines(true);
renderer.setUseFillPaint(true);
renderer.setFillPaint(Color.white);
return chart;
}

public static JPanel createDemoPanel() {
JFreeChart chart = createChart(createDataset)();
return new ChartPanel(chart);
}

public static void main(String[] args) {
distribution demo = new distribution("distribution de Weibull");
demo.pack();
RefineryUtilities.centerFrameOnScreen(demo);
demo.setVisible(true);

```


}
}