

Simulation de la propagation d'un feu de forêt

Cette ressource est issue d'un projet de l'année de pré-master en Sciences pour l'Ingénieur de l'ENS Paris-Saclay, lié à l'enseignement d'identification paramétrique et optimisation de Javier Ojeda, Maître de conférences au département EEA. Cet enseignement s'inscrit complètement dans la logique pluridisciplinaire de la formation dans la mesure où il touche un grand nombre de domaines. Lors de ce module, un projet au choix est à réaliser mettant en application des méthodes d'identification paramétrique ou d'optimisation. Élise Foulatier ayant déjà été sensibilisée aux risques d'incendie lors de périodes de sécheresse, a décidé de s'intéresser à la prévision de la propagation d'un feu de forêt. L'enseignement présente différentes méthodes d'estimation paramétrique ainsi que les cadres dans lesquelles elles s'appliquent. Plusieurs de ces méthodes ont été appliquées selon les besoins présents au cours du projet.

Les recherches sur la propagation des feux de forêt ont montré que les connaissances scientifiques devaient être importantes et variées pour expliquer les différents phénomènes. Aussi, ce projet a été soumis à un grand nombre d'hypothèses et comporte ses limites vis-à-vis des phénomènes réels. Il se veut avant tout être une application des méthodes vues en enseignement et non une simulation de phénomènes physiques bien précis de propagation du feu (comme la convection par exemple).

Cette ressource présente, après la réalisation de la simulation d'un incendie, la démarche mise en place afin d'estimer l'effet de la vitesse du vent et de son orientation sur la probabilité de propagation du feu, puis l'exploitation des mesures permettant d'estimer un modèle de probabilités au sens des moindres carrés.

1 – Introduction

Le réchauffement climatique est à l'origine d'un grand nombre de phénomènes climatiques catastrophes. Les sécheresses sont de plus en plus nombreuses et provoquent entre autres des feux de forêt aux conséquences désastreuses. Afin de prévoir les zones d'intervention optimales des équipes de pompier, il est important de pouvoir prédire la manière dont va se propager le feu. L'enjeu du projet est de **prévoir quelles vont être les zones atteintes par le feu** en fonction de certains paramètres.

On peut distinguer trois grands types de facteurs influençant la propagation d'un feu :

- **La densité et la distribution de végétaux** : en particulier leur teneur en eau est intéressante ;
- **Facteurs topographiques** : relief, pente, exposition des versants ;
- **Facteurs météorologiques** : pluviométrie, température, hygrométrie, vents.

Un incendie résultant de plusieurs phénomènes physiques complexes, aucune étude n'a permis à ce jour d'aboutir à une maîtrise complète des concepts en jeu. Les expériences menées sur ce sujet sont peu nombreuses et il existe quelques logiciels simulant l'évolution des feux de forêt (comme *Farsite* ou *FireTactic*). Ces logiciels ayant pour but d'aider des professionnels, ils ne sont pas accessibles facilement et n'ont pas pu être utilisés dans le cadre de ce projet.

Il est donc évident que le projet mené comportera un grand nombre de simplifications par rapport aux phénomènes réels. Les lois physiques se cachant derrière un certain nombre de paramètres étant relativement complexes, les paramètres les plus abordables à notre niveau sont la **vitesse du vent** (et sa direction) ainsi que le **relief de la zone boisée** considérée. Étant donnée la difficulté d'établir une loi mettant en jeu des phénomènes physiques différents simultanément, on choisit de ne se concentrer que sur l'un des deux facteurs cités. D'après le sujet de thèse d'Alexis Marchand portant sur la propagation des feux de végétation (page 12 de la référence [1]), l'effet du vent est le paramètre le plus influent sur la propagation d'un incendie.

Les facteurs d'intérêt sont donc les vitesses du vent selon les deux directions principales. L'orientation du vent se déduit automatiquement à partir des vitesses selon les deux directions principales. On considérera tout au long du projet une zone boisée plane avec une végétation homogène et sans obstacle.

La démarche adoptée pour mener à bien le projet sera la suivante. Dans un premier temps, il s'agit de mettre en place une **simulation** de l'incendie avec des paramètres régissant la propagation fixés arbitrairement. Puis, on va **estimer ces paramètres pour différentes vitesses du vent**. Ces estimations constituent des mesures exploitables en vue de finalement **déterminer une loi** décrivant la probabilité de contamination d'une case saine avoisinant une case brûlante en fonction de la vitesse du vent.

Les codes utiles au déroulement du projet ont été écrits en langage *Python* et sont téléchargeables en [annexe](#) de ce document. Des vidéos présentant certaines simulations et pouvant faciliter la compréhension sont également disponibles.

2 – Réalisation d'une simulation d'incendie

On s'intéresse dans cette partie à la simulation d'un incendie. On va d'abord expliquer quelles grandeurs ont été choisies pour réaliser la simulation puis on étudiera l'algorithme permettant de réaliser la simulation. Le code associé se nomme *1_Simulation_incendie*.

On décrit dans un premier temps l'espace concerné par l'incendie. Cet espace est quadrillé et représenté par des cases. On considère une zone carrée dont le côté est de longueur L (égale à 1 kilomètre dans tout le projet). On définit une variable *longueur* désignant l'intervalle entre deux cases voisines. Si *longueur* est faible, le maillage de l'espace est fin. Pour que la simulation soit assez rapide, on prend *longueur* = 50 m pour le moment mais cette variable peut être amenée à évoluer par la suite. Pour modéliser l'espace, on génère une matrice carrée de taille $L \times L$ comportant $N \times N$ éléments (avec $N = \frac{L}{longueur} + 1$).

On considère que chaque case de l'espace ne peut être que dans l'un des trois états suivants, états auxquels sont associées des couleurs qui seront celles utilisées pour visualiser la propagation de l'incendie :

- Case saine : elle n'a pas été atteinte par le feu, on désignera ces cases comme des cases vertes par la suite et le coefficient correspondant dans la matrice est égal à 0 ;
- Case brûlante : elle est atteinte par le feu qui n'est pas éteint à cet endroit, on désignera ces cases comme des cases rouges dont le coefficient correspondant dans la matrice est égal à 1 ;
- Case brûlée : elle a été atteinte par le feu mais celui-ci est désormais éteint à cet endroit, on désignera ces cases comme des cases noires dont le coefficient correspondant dans la matrice est égal à 2.

Pour simuler l'incendie, on considère deux probabilités p_1 et p_2 . Ces probabilités sont très importantes pour pouvoir simuler la propagation du feu. p_1 correspond à la probabilité qu'une case verte voisine d'une case rouge soit contaminée par le feu (c'est-à-dire qu'elle devienne rouge à l'instant suivant). Elle dépend donc de la **vitesse du vent**. p_2 correspond à la probabilité qu'une case rouge devienne noire à l'instant suivant. Pour cette première simulation, on a pris $p_1 = 0,5$ et $p_2 = 0,3$. Il faut également spécifier la position initiale du feu.

On peut ensuite procéder à la simulation de la propagation du feu dans tout l'espace. On met en place une boucle "Tant que" dont le critère d'arrêt est une liste de cases rouges vide. À l'intérieur de cette boucle, on considère pour chaque case rouge la propagation du feu dans les quatre directions de l'espace. De plus, pour chaque case rouge, on étudie la possibilité qu'elle devienne noire. Le modèle est donc à **temps discret** : on considère un nouvel instant à chaque itération de la boucle.

Voici l'algorithme en langage naturel de la boucle de simulation (cf Figure 1) :

Algorithm 1 Algorithme en langage naturel de la boucle de simulation d'un incendie

```

while il y a des cases rouges : do
  for caseRouge dans listeCasesRouges : do
    xCaseRouge ← abscisse de caseRouge
    yCaseRouge ← ordonnée de caseRouge
    for chaque direction de propagation do
      if extrémité correspondante non atteinte then
        nombre_aleat_case_voisine ← entier aléatoire dans (1,100)
        if nombre_aleat_case_voisine <  $p_1 \times 100$  then
          case voisine ← verte
        end if
      end if
    end for
    nombre_aleat_reste_brulant ← entier aléatoire dans (1,100)
    if nombre_aleat_reste_brulant <  $p_2 \times 100$  then
      caseRouge ← noire
    end if
  end for
end while

```

Figure 1 : Algorithme en langage naturel de la simulation d'un incendie

Jusqu'à présent, pour modéliser l'incendie on a travaillé avec une matrice dont les coefficients peuvent prendre trois valeurs (0, 1 ou 2). Pour mieux observer le phénomène de propagation, on utilise la librairie *matplotlib.pyplot* de Python et on affiche à chaque itération de la boucle les cases colorées modélisant l'espace. La vidéo 1 accessible en [annexe](#) présente une première simulation affichée avec *matplotlib.pyplot*.

On dispose désormais d'une simulation d'un incendie. Celle-ci va nous permettre d'obtenir des mesures de la valeur de p_1 pour différentes vitesses du vent.

3 – Démarche expérimentale : obtention des mesures

Comme dit en introduction, il n'est pas possible dans le cadre du projet d'obtenir des mesures expérimentales, que ce soit à partir d'expérimentations personnelles ou à partir d'une base de données. On va donc réaliser plusieurs simulations en s'appuyant sur le travail décrit précédemment. Pour chacune des simulations, on estimera la valeur de p_1 . On obtient ainsi plusieurs valeurs de p_1 pour chacune des vitesses choisies.

Dans cette partie, on va travailler avec un vent d'orientation connue (purement horizontale ou purement verticale pour que le problème soit plus simple à traiter) et dont la vitesse est variable. Le vent modifie la vitesse de propagation du feu dans une direction donnée. C'est donc la probabilité p_1 uniquement qui est impactée par la présence du vent. On réalise des estimées de p_1 pour

différentes vitesses du vent et à partir de ces "mesures" on pourra établir une loi de p_1 (cf section 4). On se référera dans cette partie aux programmes *2_Experimentation_vent_ouest_est* et *3_Récupération_données*.

3.1 - Estimation du sens du maximum de vraisemblance de p_1

Dans cette sous-partie, on montre comment obtenir une "mesure" de p_1 en effectuant une **estimée au sens du maximum de vraisemblance**.

À chaque itération, on observe notamment le nombre de cases vertes et le nombre de nouvelles cases rouges.

Pour expliquer notre démarche, on va utiliser l'exemple suivant :

- à $k=2$: il y a 350 cases vertes et 16 nouvelles cases rouges ;
- à $k=3$: il y a 325 cases vertes et 25 nouvelles cases rouges ;
- à $k=4$: il y a 285 cases vertes et 40 nouvelles cases rouges.

La probabilité p_1 correspondant à la probabilité qu'une case verte devienne rouge, on la nomme dans cette partie $p_{R|V}$ pour faciliter la compréhension. Pour chaque case, on considère une suite de valeurs aléatoires X_k qui peuvent prendre trois valeurs dans $\{V,R,N\}$ correspondant à l'état de la case. On cherche à estimer $p_{R|V} = P(X_{k+1} = R | X_k = V)$.

L'objectif est de déterminer la fonction de vraisemblance associée à $p_{R|V}$. On introduit la variable aléatoire $V_{k+1|k}$ qui donne le nombre de cases vertes à l'itération $k+1$. On introduit aussi la variable aléatoire $RV_{k+1|k}$ donnant le nombre de cases nouvellement rouges à l'itération $k+1$ et la variable aléatoire VA_{k+1} donnant le nombre de cases vertes atteignables par le feu à l'itération $k+1$ (c'est-à-dire les cases voisines d'une case rouge qui ne sont pas rouges elles-mêmes). On introduit la **fonction de vraisemblance** $j(p_{R|V})$ appliquée à l'exemple cité précédemment :

$$j(p_{R|V}) = P(V_{4|3} = 285, RV_{4|3} = 40, V_{3|2} = 25, V_{2|1} = 350, RV_{2|1} = 16 | p_{R|V})$$

D'où, en appliquant successivement la propriété des probabilités composées (avec C qui est une constante indépendante de $p_{R|V}$:

$$j(p_{R|V}) = C \times P(V_{4|3} = 285, RV_{4|3} = 40 | p_{R|V}) \times P(V_{3|2} = 25, RV_{3|2} = 25 | p_{R|V}) \times P(V_{2|1} = 350, RV_{2|1} = 16 | p_{R|V})$$

Or, $P(V_{2|1} = 350, RV_{2|1} = 16 | p_{R|V})$ correspond à la probabilité que 350 cases soient vertes et que 16 cases soient devenues rouges à l'itération $k = 2$. Une case verte à l'itération k peut être soit verte soit rouge à l'itération suivante et ce avec une probabilité $p_{R|V}$. La variable aléatoire correspondante suit donc une **loi de Bernoulli** avec une probabilité de succès (passage à l'état rouge) qui vaut $p_{R|V}$. Si on considère l'ensemble des cases de l'espace, la variable aléatoire associée à la probabilité $P(V_{2|1} = 350, RV_{2|1} = 16 | p_{R|V})$ suit donc une **loi binomiale de paramètres** $n = VA_1$ et $p = p_{R|V}$, avec VA_1 le nombre de points verts atteignables à l'itération 1.

$$\text{On a ainsi : } P(V_{2|1} = 350, RV_{2|1} = 16 | p_{R|V}) = \binom{VA_1}{RV_{2|1}} p_{R|V}^{RV_{2|1}} (1 - p_{R|V})^{VA_1 - RV_{2|1}}$$

Et donc : $j(p_{R|V}) = C \times \prod_{k=1}^N \binom{VA_k}{RV_{k+1|k}} p_{R|V}^{RV_{k+1|k}} (1 - p_{R|V})^{VA_k - RV_{k+1|k}}$, en considérant tous les instants de la simulation.

Cette fonction de vraisemblance n'est pas simple à maximiser. La fonction de logvraisemblance est plus aisée à exploiter et atteint son maximum au même point que $j(p_{R|V})$.

$$\log(j(p_{R|V})) = \log C + \sum_{k=1}^N \log\left(\frac{VA_k}{RV_{k+1|k}}\right) + RV_{k+1|k} \log(p_{R|V}) + (VA_k - RV_{k+1|k}) \log(1 - p_{R|V})$$

Donc :

$$\frac{\partial \log(j(p_{R|V}))}{\partial p_{R|V}} = \sum_{k=1}^N \frac{RV_{k+1|k}}{p_{R|V}} - \frac{VA_k - RV_{k+1|k}}{1 - p_{R|V}}$$

L'estimée au sens du maximum de vraisemblance étant l'argument qui maximise $\log(j(p_{R|V}))$:

$$\frac{\partial \log(j(\hat{p}_{R|V}))}{\partial p_{R|V}} = 0 \Rightarrow \hat{p}_{R|V} = \frac{\sum_{k=1}^N RV_{k+1|k}}{\sum_{k=1}^N VA_k}$$

À partir de cette dernière expression, on peut estimer au sens du maximum de vraisemblance la probabilité $p_1 = p_{R|V}$ à la fin de chaque simulation. En réalisant quelques simulations avec une probabilité p_1 connue, on vérifie qu'on obtient des résultats satisfaisants. On remarque que si le maillage est trop faible, il y a une grande incertitude sur l'estimée de p_1 . Ceci est cohérent car l'estimée est toujours plus précise si on considère un grand nombre de points. On veillera donc lors de la réalisation des mesures à avoir un maillage assez fin de l'espace.

3.2 - Introduction de l'effet du vent dans la simulation

Maintenant qu'on est en mesure d'estimer p_1 au sens du maximum de vraisemblance, on peut obtenir des mesures de p_1 pour **différentes vitesses du vent**. On considère un vent de direction purement horizontale allant d'Ouest en Est pour que le problème soit plus simple à traiter. Si on se réfère à l'algorithme de simulation écrit en langage naturel dans la section 2, on considère que la probabilité que le feu se propage est la même dans les quatre directions. Dans le cas présent, on veut modéliser l'effet d'un vent de vitesse V_x qui irait d'Ouest en Est. Il semble donc évident que le feu se propagera davantage dans la direction des x croissants que dans les autres. On introduit **quatre facteurs** (un pour chaque direction) qui vont augmenter ou diminuer la probabilité que le feu se propage au point voisin **dans une direction donnée**. Ceci permet de "favoriser" la propagation du feu dans une direction particulière.

Dans le code fourni (*2_Experimentation_vent_ouest_est*), ces quatre facteurs sont nommés :

- *facteur_propagation_vx_plus* : pour la direction des x croissants (d'Ouest en Est) ;
- *facteur_propagation_vx_moins* : pour la direction des x décroissants (d'Est en Ouest) ;
- *facteur_propagation_vy_plus* : pour la direction des y croissants (du Sud vers le Nord) ;
- *facteur_propagation_vy_moins* : pour la direction des y décroissants (du Nord vers le Sud).

De façon évidente, les quatre facteurs sont égaux à 1 en l'absence de vent. Comme le vent est ici **unidirectionnel**, on ne modifie que le facteur correspondant à la propagation dans le sens du vent (ici il s'agit du facteur *facteur_propagation_vx_plus*). On pourra envisager par la suite de diminuer le facteur correspondant à la propagation dans le sens des x décroissants.

Cherchons à déterminer une loi de *facteur_propagation_vx_plus* dépendant de V_x . On suppose que les facteurs suivent une **loi affine**. S'il n'y a pas de vent, la probabilité que le feu contamine une case voisine est inchangée. Donc *facteur_propagation_vx_plus*($V_x = 0$) = 1. On suppose maintenant qu'à partir d'une certaine vitesse de vent (notée V_{max}), le feu se propage forcément au point voisin. On a donc :

$$facteur_propagation_vx_plus(V_x = V_{max}) = \frac{1}{p_1}.$$

À partir de ces deux valeurs on obtient la loi du facteur dans la direction des x croissants :

$$\text{facteur_propagation_vx_plus}(V_x) = \frac{V_x}{V_{max}} \frac{1-p_1}{p_1} + 1$$

Si on se contente uniquement de modifier le facteur dans le sens des x croissants, la modélisation de l'incendie est assez éloignée de la réalité. En effet, le feu se propage dans le sens opposé à celui du vent de la même manière qu'il y ait du vent ou non. Il faut donc trouver une loi affine de $\text{facteur_propagation_vx_moins}$. On dispose de deux valeurs caractéristiques de $\text{facteur_propagation_vx_moins}$:

$$\begin{cases} \text{facteur_propagation_vx_moins}(V_x = 0) = 1 \\ \text{facteur_propagation_vx_moins}(V_x = V_{max}) = 0 \end{cases}$$

En effet, on suppose qu'à partir d'une certaine vitesse de vent le feu ne peut pas se propager dans la direction opposée au vent. On a alors la loi du facteur dans la direction des x décroissants :

$$\text{facteur_propagation_vx_moins}(V_x) = 1 - \frac{V_x}{V_{max}}$$

Cette modélisation de la propagation du feu est cohérente : on considère souvent qu'en terrain plat, la propagation du feu s'effectue de manière elliptique (cf Figure 2). On observe bien sur cette même figure que le feu ne se propage pas dans la direction opposée à la flèche bleue, indiquant le sens du vent.

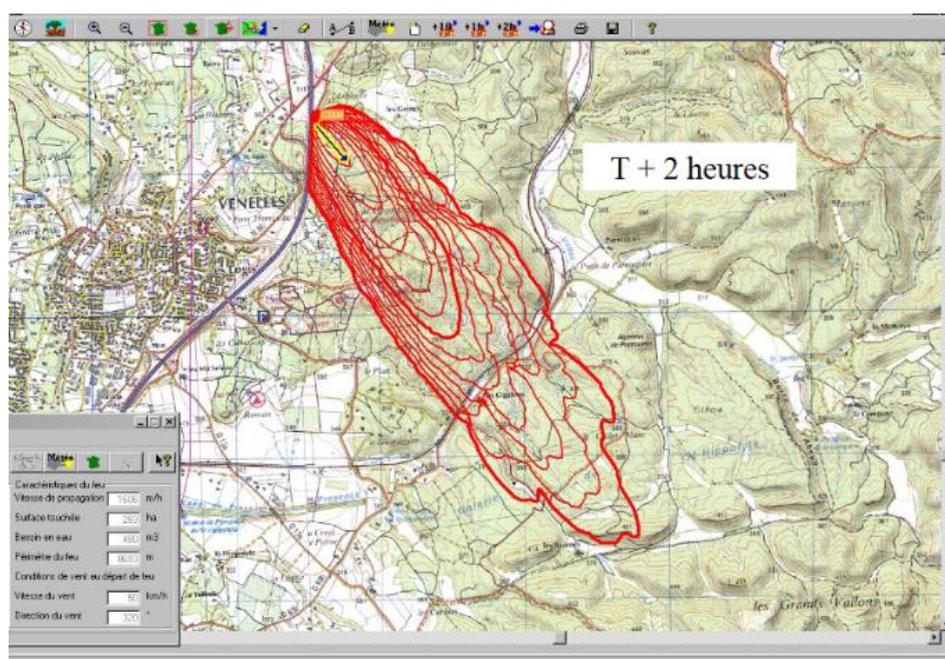


Figure 2 : Simulation de la propagation d'un feu de forêt à l'aide du logiciel FireTactic

La simulation de l'incendie doit être légèrement modifiée en vue d'obtenir cette forme elliptique. Observons tout d'abord la manière dont se propage le feu dans l'état actuel de la simulation (cf Vidéo 2 en [annexe](#)). On voit bien que le feu se propage plus vite dans le sens des x croissants. Le front du feu a une forme plutôt proche d'une ellipse avant d'atteindre l'extrémité du quadrillage. Cependant, la simulation continue une fois l'extrémité Est atteinte et la zone brûlée n'a pas la forme d'une ellipse en fin de simulation. Dans la situation modélisée, on a pris un échantillon de zone boisée (on suppose qu'il y a encore de la végétation en-dehors de l'espace représenté) et on ne s'intéresse pas aux conditions aux limites qui sont délicates à traiter. On considère donc qu'on arrête la simulation quand le côté Est du quadrillage est atteint. On obtient alors la simulation visible en Vidéo. La forme de la surface boisée brûlée est alors la suivante (cf Figure 3). Sa forme approche celle d'une ellipse.

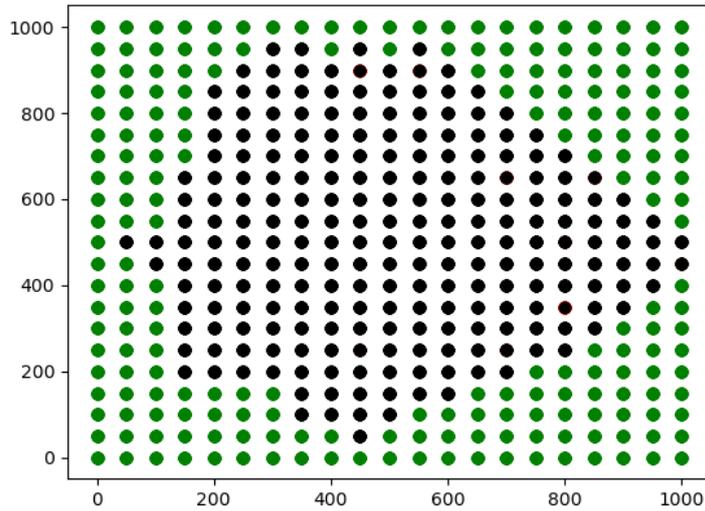


Figure 3 : Allure de la surface brûlée après modification du code

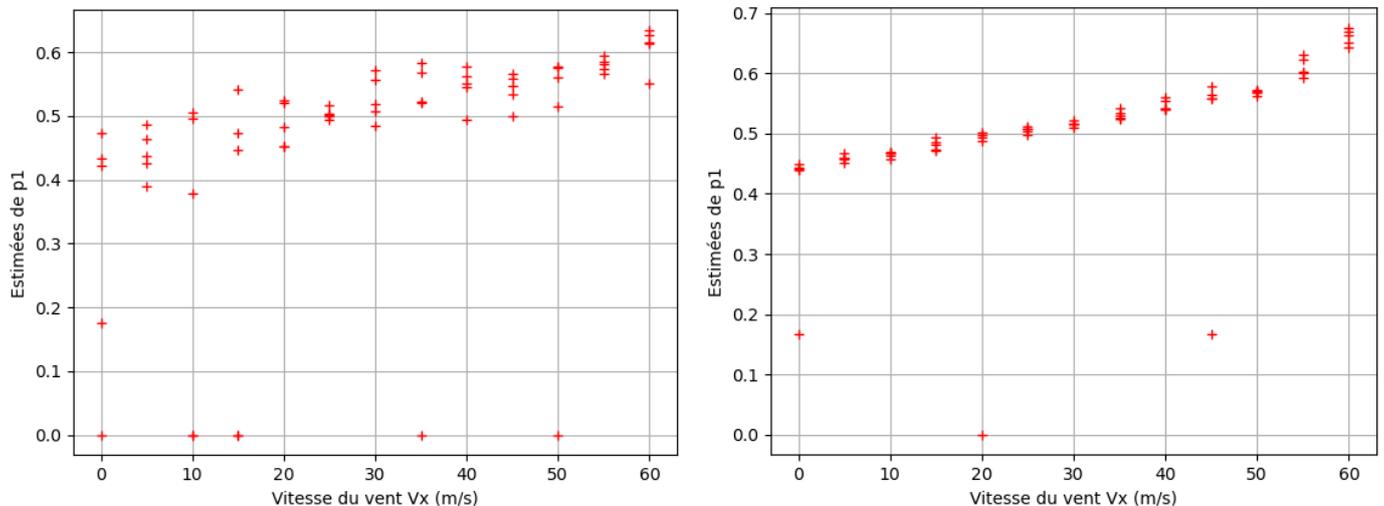
Pour réaliser cette simulation, on a légèrement modifié l'algorithme présenté en section 2. On a introduit une variable booléenne j , à laquelle on affecte la valeur *Faux* en début d'algorithme. Si la case rouge étudiée a pour abscisse L (elle est située sur le bord Est du quadrillage), on affecte *Vrai* à j , ce qui permet de stopper la propagation du feu. La simulation étudie toujours le passage de l'état rouge à noir des cases rouges restantes.

3.3 - Démarche expérimentale pour obtenir des mesures de p_1

Grâce aux deux sous-parties précédentes, il ne reste plus qu'à obtenir plusieurs estimées de p_1 pour des vitesses V_x différentes. Le code correspondant à cette sous-partie est *3_Récupération_données*.

On considère comme dans la partie précédente un vent allant d'Ouest en Est. Pour chaque vitesse, on calcule cinq estimées de p_1 . On prend des vitesses comprises entre 0 et 60 km/h, espacées de 5 km/h. Avec le même maillage de l'espace que précédemment (*longueur* = 50 m), on obtient les mesures présentées en Figure 4.a. Les mesures sont **fortement bruitées** et une loi de p_1 ne se dégage pas clairement.

On décide donc d'affiner le maillage en prenant *longueur* = 10 m. On obtient alors les mesures présentées en Figure 4.b.



(a) Longueur = 50 m

(b) Longueur = 10 m

Figure 4 : Estimées au sens du maximum de vraisemblance de p_1 en fonction de la vitesse du vent pour différents maillages

Des mesures aberrantes subsistent : elles concernent les points très éloignés des autres points de mesure. Les autres points de mesures sont beaucoup moins dispersés avec *longueur* = 10 m donc on travaillera avec le nuage de points obtenus en Figure 4.b. pour avoir un modèle de p_1 satisfaisant.

Dans cette partie, on a montré comment obtenir des mesures à l'aide d'une estimée au sens du maximum de vraisemblance alors que des expérimentations étaient impossibles et qu'on ne disposait pas de base de données. On va maintenant passer à la dernière phase du projet qui consiste à trouver une loi mathématique de p_1 en fonction de la vitesse du vent.

4 – Estimation d'un modèle mathématique de la probabilité de la propagation du feu

L'estimation paramétrique se trouve dans le code *4_Estimation_moindres_carres*. La liste des points présente en début de programme a été récupérée grâce à la méthode décrite dans la section 3.

On va réaliser une estimée au sens des moindres carrés. En effet, un certain nombre de critères pour l'estimation paramétrique ne sont pas envisageables comme on n'a aucune information sur la nature du bruit de mesure ni de connaissances a priori sur le vecteur des paramètres du modèle. Pour utiliser une méthode des moindres carrés, il faut au préalable définir un modèle mathématique.

D'après la Figure 4.b, on peut envisager un modèle exponentiel de la forme $y_m(v) = \lambda e^{\alpha v}$ ou un modèle polynomial d'ordre 1, 2 ou 3.

On se concentre d'abord sur un modèle exponentiel. Ce modèle de comportement n'est pas linéaire en les paramètres. On peut le rendre linéaire en appliquant la fonction logarithme népérien à l'expression de y_m . On a alors : $\widetilde{y}_m = \ln(\lambda) + \alpha v = (1 \ v) \cdot \underline{q}$

avec $\underline{q}^T = (\ln(\lambda), \alpha)$.

On réalise une estimée au sens des moindres carrés de \underline{q} mais l'estimée de \underline{q} ne correspond pas forcément à l'estimée au sens des moindres carrés de \underline{p} .

On pose \underline{y} , le vecteur donnant les estimées de p_1 pour différentes vitesses et $\widetilde{\underline{y}}$ le vecteur correspondant auquel on a appliqué la fonction logarithme népérien. On cherche à minimiser le critère suivant :

$$C_{MC}(\underline{q}) = (\widetilde{\underline{y}} - \widetilde{\underline{y}}_m(\underline{q}))^T \cdot (\widetilde{\underline{y}} - \widetilde{\underline{y}}_m(\underline{q}))$$

Comme le modèle est linéaire en les paramètres on peut écrire $\widetilde{\underline{y}}_m(\underline{q})$ en fonction de la matrice des régresseurs \underline{R} et du vecteur des paramètres \underline{q} :

$$\widetilde{\underline{y}}_m(\underline{q}) = \underline{R} \cdot \underline{q} \quad \text{avec} \quad \underline{R} = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ \cdot & \cdot \\ \cdot & \cdot \\ 1 & 60 \\ 1 & 60 \end{pmatrix}$$

On peut ainsi développer le critère des moindres carrés :

$$C_{MC}(\underline{q}) = \widetilde{\underline{y}}^T \cdot \widetilde{\underline{y}} - 2\widetilde{\underline{y}}^T \underline{R} \cdot \underline{q} + \underline{q}^T \cdot \underline{R}^T \cdot \underline{R} \cdot \underline{q}$$

En dérivant par rapport à \underline{q} , on obtient l'estimée \hat{q}_{MC} au sens des moindres carrés :

$$\hat{q}_{MC} = (\underline{R}^T \cdot \underline{R})^{-1} \cdot \underline{R}^T \cdot \underline{\tilde{y}}$$

On déduit ensuite une première approximation du vecteur des paramètres \underline{p} :

$$\hat{\underline{p}} = (e^{q_{MC}(1)}, q_{MC}(2))^T$$

Avec une telle approche on obtient le **modèle présenté en Figure 5.a.**

On constate qu'il y a des **mesures aberrantes** et on peut leur donner un **poids faible** en utilisant la méthode des **moindres carrés pondérés**. Le critère est alors de la forme :

$$C_{MC}(q) = (\underline{\tilde{y}} - \underline{\tilde{y}_m}(q))^T \cdot \underline{W} \cdot (\underline{\tilde{y}} - \underline{\tilde{y}_m}(q))$$

En développant le calcul on obtient :

$$C_{MC}(q) = \underline{\tilde{y}}^T \cdot \underline{W} \cdot \underline{\tilde{y}} - \underline{\tilde{y}}^T \cdot \underline{W} \cdot \underline{R} \cdot \underline{q} - \underline{q}^T \cdot \underline{R}^T \cdot \underline{W} \cdot \underline{\tilde{y}} + \underline{q}^T \cdot \underline{R}^T \cdot \underline{W} \cdot \underline{R} \cdot \underline{q}$$

Ce qui donne en dérivant (comme \underline{W} est une matrice diagonale donc symétrique) :

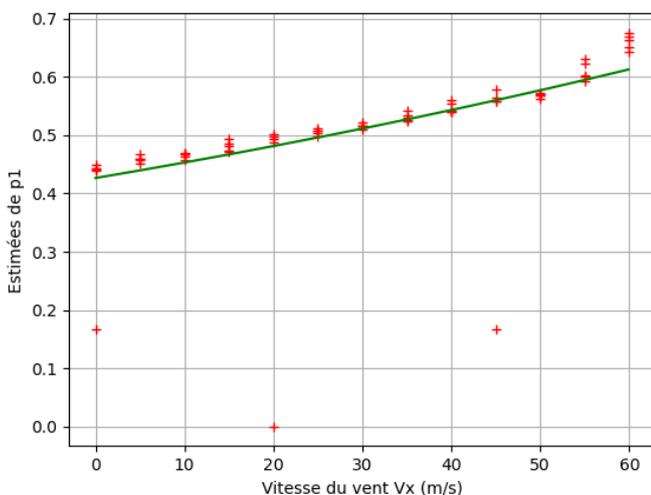
$$\frac{\partial C_{MCP}(q)}{\partial \underline{q}} = -2 \cdot \underline{R}^T \cdot \underline{W} \cdot \underline{\tilde{y}} + 2 \cdot \underline{R}^T \cdot \underline{W} \cdot \underline{R} \cdot \underline{q}$$

Donc l'estimée au sens des moindres carrés pondérés a pour expression :

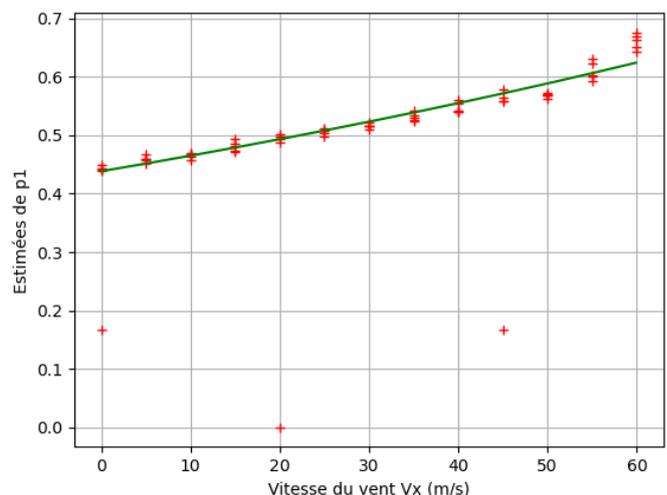
$$\hat{q}_{MCP} = (\underline{R}^T \cdot \underline{W} \cdot \underline{R})^{-1} \cdot \underline{R}^T \cdot \underline{W} \cdot \underline{\tilde{y}}$$

Si le **coefficient diagonal w_i** est élevé, il va donner un **poids important à la mesure**. On définit donc une matrice ne comportant que des 1 sur sa diagonale. On parcourt la liste des mesures : quand une mesure est trop éloignée des autres, on égalise le coefficient correspondant w_i à **0,001**.

En procédant de cette manière on obtient la loi de p_1 présentée en Figure 5.b.



(a) Sans pondération



(b) Avec pondération

Figure 5 : Estimées de la loi exponentielle de p_1 au sens des moindres carrés

On estime de la même manière un **modèle polynomial** afin de le comparer au modèle exponentiel. L'avantage de ce modèle est qu'il est directement **linéaire en les paramètres**. On obtient donc de manière exacte l'estimée au sens des moindres carrés du vecteur des paramètres \underline{p} .

On construit la matrice des régresseurs $\underline{\underline{R}}$ correspondant au modèle. Sa taille dépend de l'ordre du polynôme. Si on choisit un polynôme d'ordre 3 on a :

$$\underline{\underline{R}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 60 & 60^2 & 60^3 \\ 1 & 60 & 60^2 & 60^3 \end{pmatrix}$$

car $y_m(v, p) = (1 \ v \ v^2 \ v^3) \cdot \underline{p}$

Comme précédemment, on a l'estimée du vecteur des paramètres \underline{p} au sens des **moindres carrés** :

$$\hat{\underline{p}}_{MC} = \left(\underline{\underline{R}}^T \cdot \underline{\underline{R}} \right)^{-1} \cdot \underline{\underline{R}}^T \cdot \underline{\tilde{y}}$$

Et au sens des moindres carrés pondérés :

$$\hat{\underline{p}}_{MCP} = \left(\underline{\underline{R}}^T \cdot \underline{\underline{W}} \cdot \underline{\underline{R}} \right)^{-1} \cdot \underline{\underline{R}}^T \cdot \underline{\underline{W}} \cdot \underline{\tilde{y}}$$

Pour des polynômes d'ordre 1, 2 et 3, on peut tracer les lois estimées de p_1 sans et avec pondération (cf Figure 6).

Pour chaque modèle, on calcule la **valeur du critère des moindres carrés**. Ceci permet d'évaluer pour quel modèle l'erreur est la plus faible.

On regroupe les valeurs des critères dans le Tableau 1.

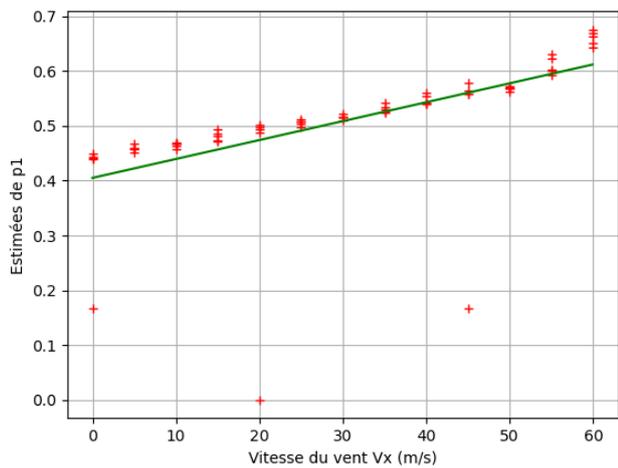
Type de modèle mathématique	Pondération	Évaluation du critère des moindres carrés
Loi exponentielle	Non	0,476
Loi exponentielle	Oui	0,0136
Polynôme d'ordre 1	Non	0,475
Polynôme d'ordre 1	Oui	0,017
Polynôme d'ordre 2	Non	0,466
Polynôme d'ordre 2	Oui	$9,35 \cdot 10^{-3}$
Polynôme d'ordre 3	Non	0,478
Polynôme d'ordre 3	Oui	0,0187

Tableau 1 : Valeurs du critère des moindres carrés pour les différentes estimées réalisées

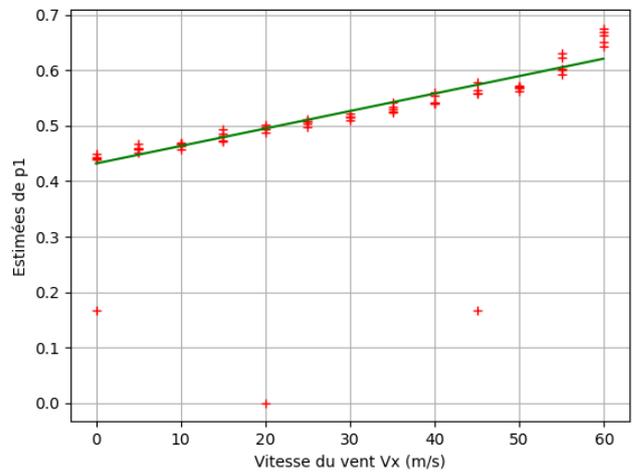
D'après le Tableau 1, la méthode des moindres carrés pondérés est toujours plus précise, ce qui était attendu. De plus, on remarque qu'on **minimise la somme des carrés des erreurs** avec un **modèle polynomial d'ordre 2**. Ce modèle de p_1 est un **modèle de comportement, globalement identifiable** comme il s'agit d'un polynôme. On a :

$$p_1(v) = \alpha_0 + \alpha_1 v + \alpha_2 v^2$$

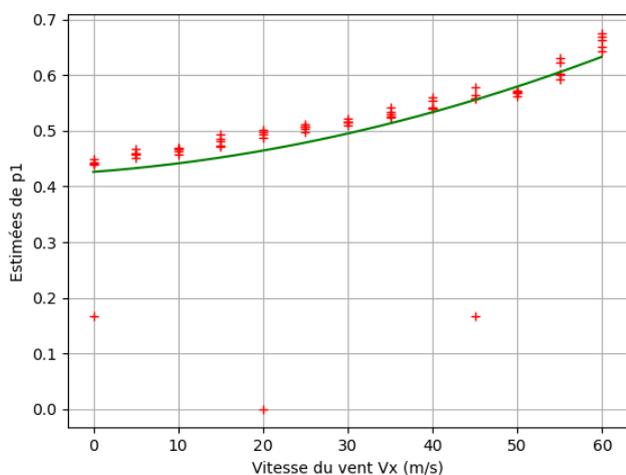
avec $\begin{cases} \alpha_0 = 0,453 \\ \alpha_1 = 9,58 \cdot 10^{-4} \text{ h/km} \\ \alpha_2 = 3,61 \cdot 10^{-5} \text{ h}^2/\text{km}^2 \end{cases}$



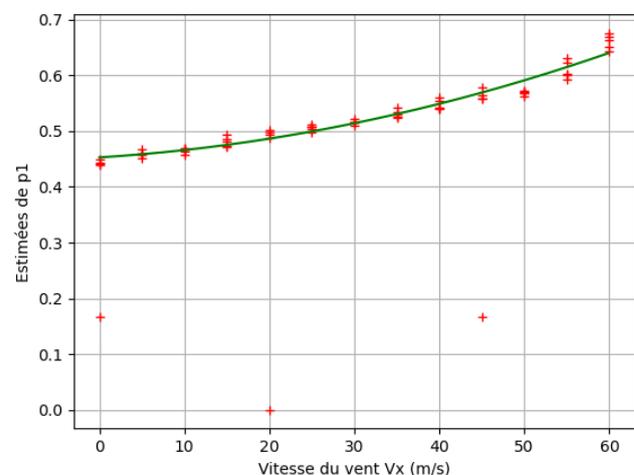
(a) Polynôme d'ordre 1 sans pondération



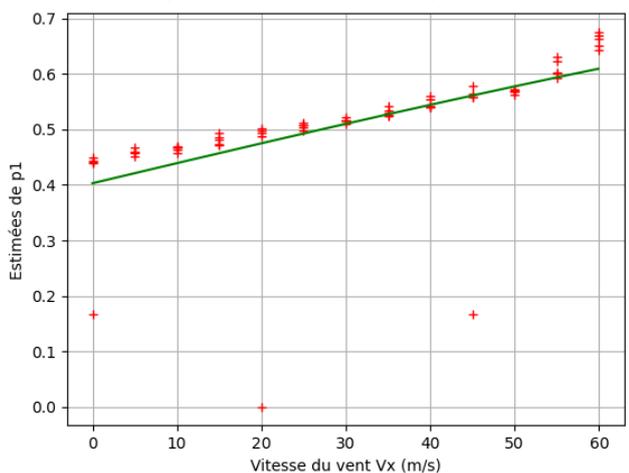
(b) Polynôme d'ordre 1 avec pondération



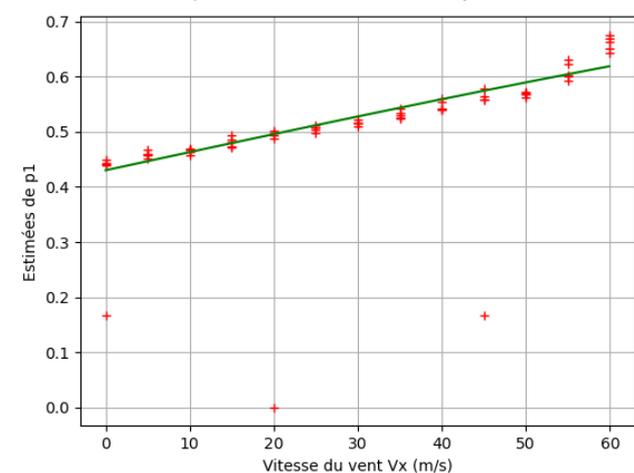
(c) Polynôme d'ordre 2 sans pondération



(d) Polynôme d'ordre 2 avec pondération



(e) Polynôme d'ordre 3 sans pondération



(f) Polynôme d'ordre 3 avec pondération

Figure 6 : Estimées des lois polynomiales de p_1 d'ordre différents obtenues par méthode des moindres carrés avec et sans pondération

5 – Conclusion

Grâce à la loi mathématique de p_1 déterminée, on a pu implémenter facilement la simulation de l'incendie pour n'importe quelle vitesse de vent (cf code `5_Simulation_finale`). Pour cela, on décompose la vitesse du vent selon les deux directions principales du repère. Si on calcule

simplement la valeur de p_1 et qu'on ne modifie pas le code de la section 2, on ne verra pas qu'une direction de propagation est privilégiée. Pour modifier ce problème, on applique donc un **coefficient** devant les probabilités utilisées pour chaque direction de telle sorte que la **somme des probabilités réellement considérées dans les deux sens d'une même direction soient égales à $2p_1$** . On ajoute également à la simulation de départ, la condition portant sur l'**arrêt de la simulation** lorsqu'une **extrémité du quadrillage est atteinte**. En opérant de la sorte, on vérifie la cohérence de la simulation en observant d'une part la **vidéo** de la simulation (cf Vidéos 4 et 5 en [annexe](#)) et **en calculant plusieurs estimées au maximum de vraisemblance de p_1** . Ces estimées sont toutes proches de la valeur de p_1 calculée à partir du modèle polynômial.

Dans le cadre de ce projet, on a mis en place une démarche permettant d'**estimer l'effet de la vitesse du vent et de son orientation** sur la probabilité p_1 de propagation du feu d'une case brûlante vers une case saine. Le modèle étant à temps discret, la simulation permet de remonter à la **vitesse de propagation du feu**. Comme dit en introduction, ce modèle ne prend comme paramètres d'entrée que les vitesses suivant les deux directions principales du repère. La modélisation du phénomène est donc **extrêmement simplifiée**. De plus, nous ne disposons pas d'informations quant à la manière dont se propage réellement le feu en fonction de la vitesse du vent ce qui nous a conduit à réaliser une **estimée au sens du maximum de vraisemblance de p_1** . Ces **mesures** sont critiquables car sans doute **éloignées de la réalité**. Par exemple, on a supposé que la probabilité p_1 (dans une direction de propagation donnée) suivait une **loi affine dépendant de la vitesse du vent** ce qui n'a **jamais été vérifié**.

Enfin, l'exploitation des mesures a permis d'estimer un modèle de p_1 au sens des **moindres carrés pondérés**. D'autres critères auraient pu être exploités mais celui des moindres carrés présente l'avantage d'être **dérivable en tout point**. Si le polynôme d'ordre 2 était le modèle minimisant la somme des carrés des erreurs **parmi les modèles mathématiques envisagés**, il demeure imparfait. On obtient avec le modèle polynomial d'ordre 2 un **coefficient de détermination linéaire $R^2 = 0,963$** . C'est un coefficient de détermination **élevé** mais avec des **modèles mathématiques plus complexes** on peut espérer augmenter ce coefficient.

S'il comporte ses limites, le travail mené ne s'est pas avéré inutile. En effet, si on dispose par le futur des **moyens** et des **connaissances requises pour obtenir des mesures réelles de p_1** en fonction de la vitesse du vent, le travail d'**estimation paramétrique** effectué dans le cadre de ce projet sera réutilisable. À la lueur de ces nouveaux résultats, un modèle mathématique plus pertinent pourra être trouvé.

6 – Annexes

Les différents codes qui ont été utilisés pour réaliser ce projet ainsi que les cinq vidéos présentant des simulations d'incendie sont téléchargeables en [annexe](#) de ce document.

Références :

[1]: Alexis Lhuillier-Marchand, Propagation des feux de végétation : expérimentations à l'échelle du laboratoire et validation d'un propagateur, Mars 2017, tel-01487015 p. 12, p.24-30

[2]: Direction Départementale des Territoires et de la Mer, Projet de Plan de Prévention des Risques Naturels d'Incendie de Forêt pour la commune du Plan-de-la-Tour, 13 octobre 2003

[3]: Diener, Méthode des moindres carrés, Université de Nice - Sophia Antipolis

Ressource publiée sur Culture Sciences de l'Ingénieur : <https://eduscol.education.fr/sti/si-ens-paris-saclay>