L’ensablement des zones urbaines et agricoles

Comment peut-on modéliser le déplacement du sable pour prévoir son avancée vers les zones urbaines et agricoles?

Numéro d’inscription 42028

Plan

I. Introduction et présentation du phénomène

II. Etude du mouvement d’un grain de sable en saltation par le PFD:

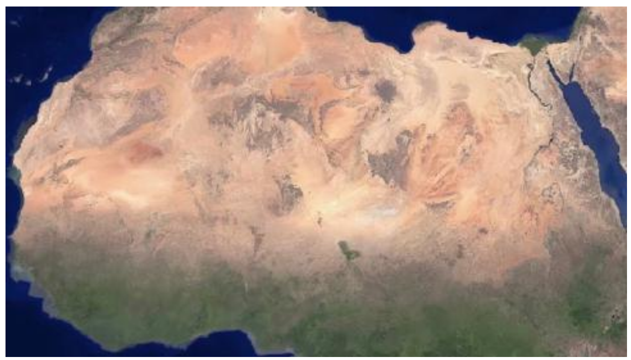
   a. **Positionnement du problème**
   b. **Équations du mouvement**
   c. **Résolution sur Python**

III. Vers un modèle plus avancé:

   a. **Introduction du modèle**
   b. **Le modèle d’Anderson**
   c. **Résolution**

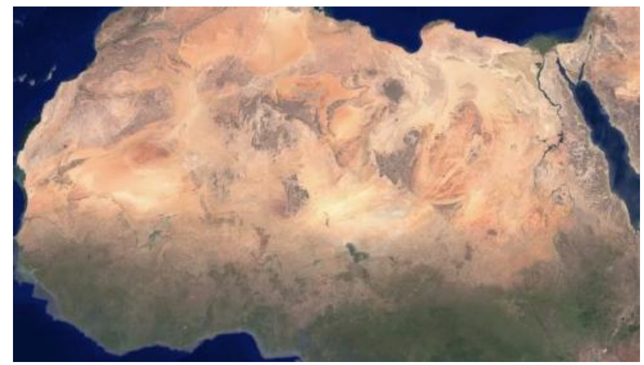
L’ensablement et sa conséquence.

- Enterrement ou recouvrement par le sable.
- Un phénomène global:



Images issu de Google Earth Timelapse

1984



Images issu de Google Earth Timelapse

2020

- Fragilisation des sols fertiles et augmentation du risque de sécheresse.

Une cause de l'ensablement

L'érosion éolienne ou transport du sable par le vent.

Mais comment se manifeste - t - elle?

La reptation

La saltation

La suspension

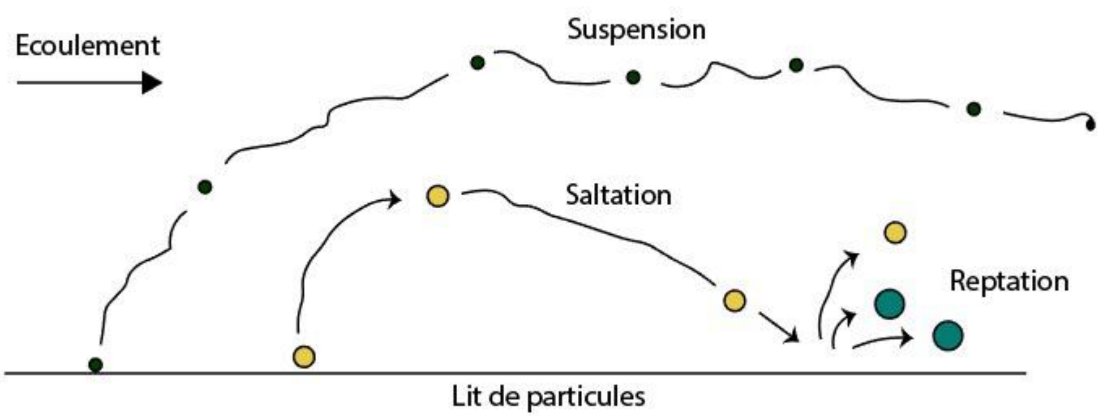


Illustration. 1 : Les différents modes de transport des grains de sable

Illustration expérimentale 1/2

J'ai étalé du sable afin d'avoir un profil le plus plan possible de longueur 31 cm. Ensuite, à l'aide d'un ventilateur, j'ai soufflé sur le sable à faible vitesse pendant environ 5 min.

Illustration expérimentale 2/2

Une bosse s'est créée à environ 18 centimètre sur l'échelle. On a pu aussi observer des grains en reptation.

Etude du mouvement d'un grain de sable en saltation par le PFD 1/5

a. Positionnement du problème

Illustration. 2 : Déclenchement de la saltation

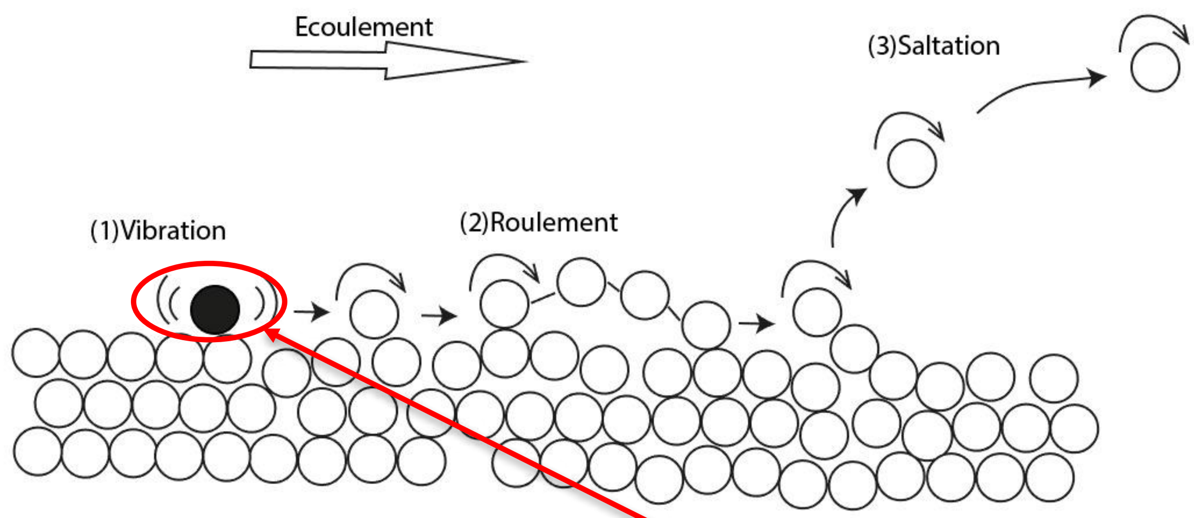
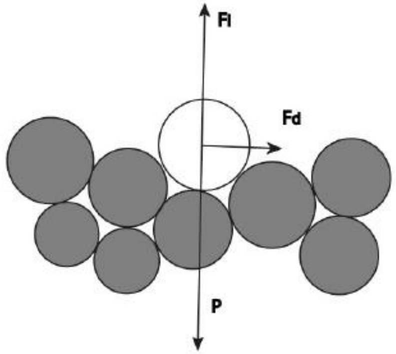


Illustration. 3 : Les forces exercé sur un grain de sable



Initiation du mouvement par le vent:

Force de portance de norme F_l

Force de trainée de norme F_d

Poids apparent de norme P

Etude du mouvement d'un grain de sable en saltation par le PFD 2/5

b. Equation du mouvement:

Force de trainée : $\overrightarrow{F_D} = -\frac{1}{4} C_D \rho_a \pi d^2 \left\|\overrightarrow{V_r}\right\| \overrightarrow{V_r}$

Force de portance : $\overrightarrow{F_L} = \frac{1}{4} \pi C_L \rho_a d^2 \left\|\overrightarrow{V_r}\right\|^2 \overrightarrow{n}$

Poids apparent : $\overrightarrow{P} = -\frac{\pi}{6} (\rho_g - \rho_a) g d^3 \overrightarrow{e_y}$

C_D : Coefficient de trainée
C_L : Coefficient de portance
$\overrightarrow{V_r}$: Vitesse relative du grain par rapport au vent.
$\vec{n}$: Normale à $\overrightarrow{V_r}$ vers le haut
d : Diamètre du grain
ρ_a : Masse volumique de l'air
ρ_g : Masse volumique grain

PFD

En considérant que:

- Grain = Sphère
- Le vent est horizontale

$$\rho_g d^3 \ddot{x} = \left(\frac{6}{4} C_D \rho_a d^2 (u - \dot{x}) - \frac{6}{4} C_L \rho_a d^2 \dot{y}\right) \sqrt{(\dot{x} - u)^2 + \dot{y}^2}$$

$$\rho_g d^3 \ddot{y} = \left(\frac{6}{4} C_L \rho_a d^2 (\dot{x} - u) - \frac{6}{4} C_D \rho_a d^2 \dot{y}\right) \sqrt{(\dot{x} - u)^2 + \dot{y}^2} - (\rho_g - \rho_a) g d^3$$

Avec u : La vitesse du vent

2 Équations non linéaire de second ordre couplé !

c. Résolution sur Python:

En utilisant Runge-Kutta pour les valeurs initiales:

V_{p0} = 1 m/s
W_{p0} = 2.5 m/s
u = 5 m/s
d = 400 µm

Que se passe-t-il quand le grain arrive au sol après son premier envol?

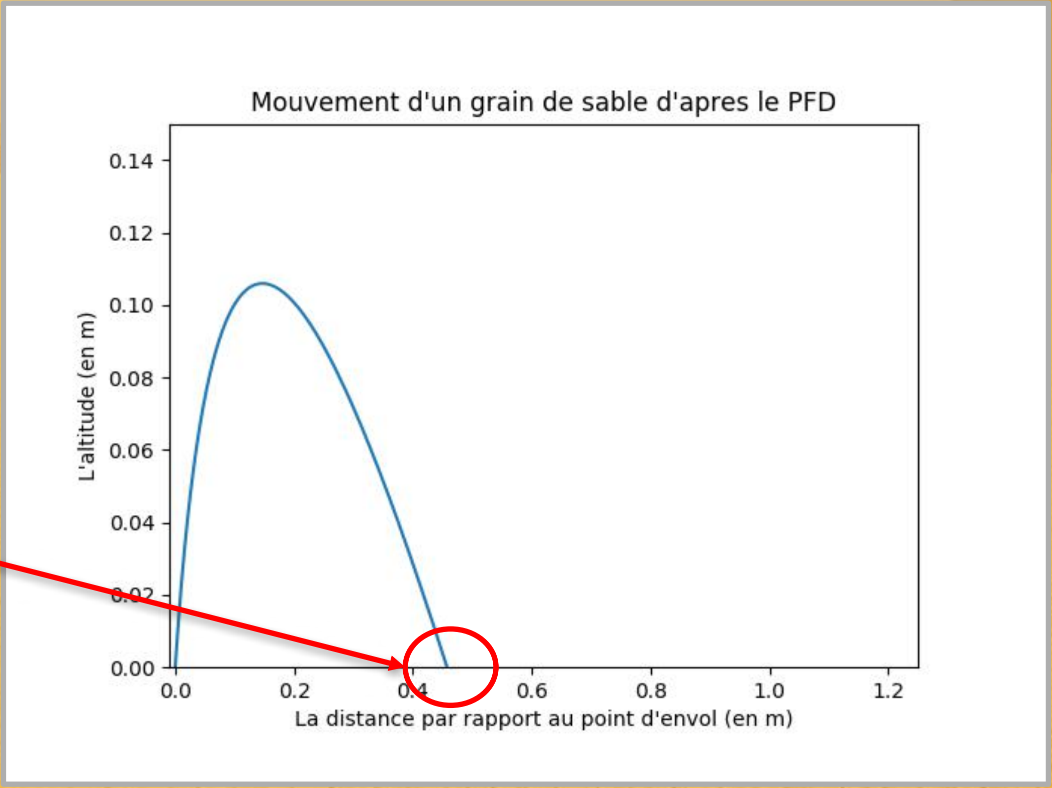


Fig.1.1

UN REBOND !

c. Résolution sur Python:

Le rebond :

Angle d'arrivé = Angle de départ

$V_{ap} = 0.6xV_{av}$

Implémentation de ces conditions dans Python :

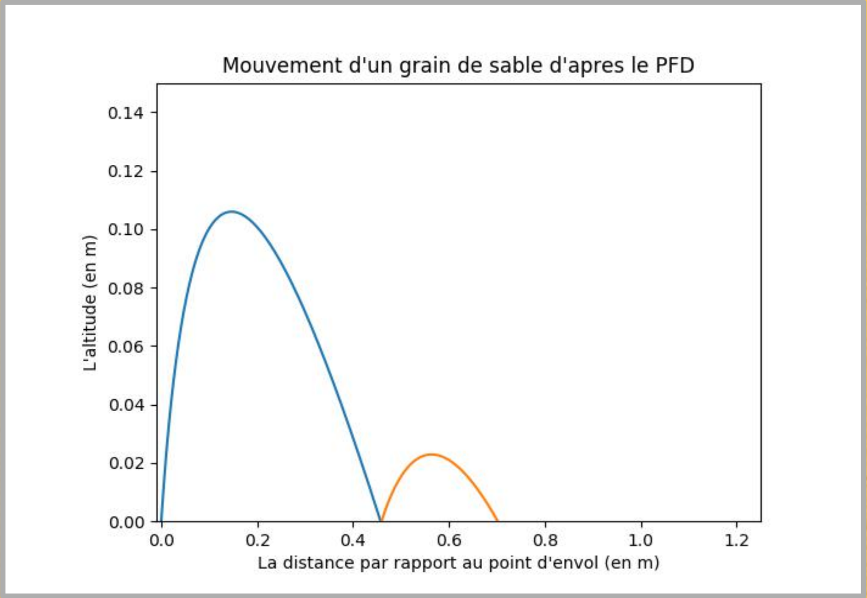


Fig. 1.2

```
La vitesse d'impact est de 3 m/s
L'angle d'incidence est 27.15 degree
```

Propriétés des grains en saltation				
Auteurs	Taille des grains	Vitesse d'impact V_i	Angle d'impact	Vitesse de rebond
White et Schulz	$350 - 710\mu m$	$160\,cm/s$	$14°$	$0{,}4\ V_i$
Willets et Rice	$150 - 600\mu m$	$400\,cm/s$	$9 - 12°$	$0,6V_i$
Napalnis, Hunt et Barrett	$100 - 200\mu m$		$11 - 14°$	

Fig.2

Etude du mouvement d'un grain de sable en saltation par le PFD 5/5

c. Résolution sur Python:

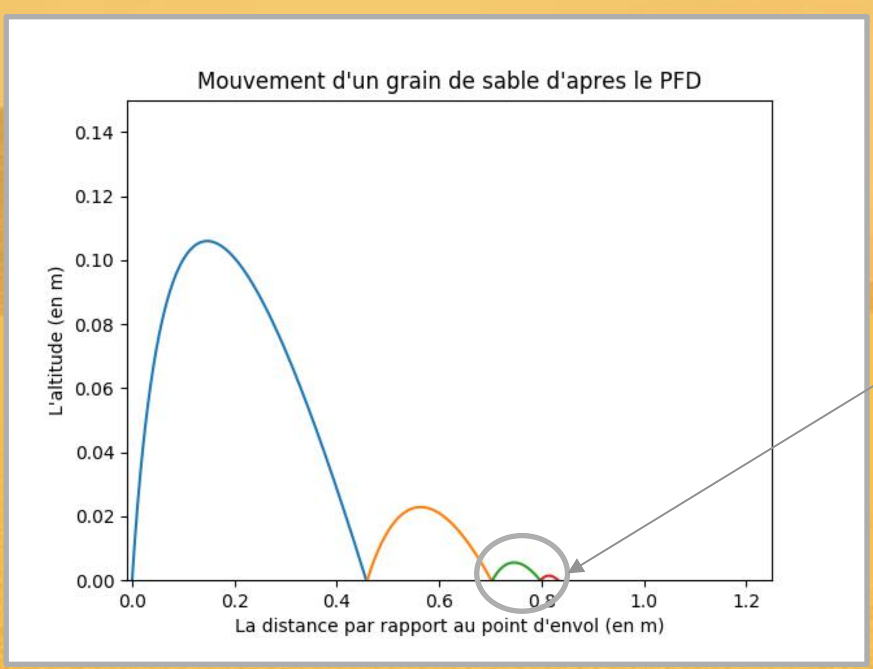


Fig. 1.3

Des sauts d'altitudes très faibles
=
REPTATION

Le phénomène de saltation ←→ Premier + Deuxième saut

Un avancement de l'ordre de 0.8 mètre

Lien de l'animation: https://youtu.be/mwA7CKGb5cM

Vers un modèle plus avancé 1/8

a. Introduction du modèle

Le vent emporte avec lui un grand nombre de grain. → Chaque grain va rebondir sur le sol → Un rebond → Grains éjecté

En considérant le profil de rides suivant: Étudions l'évolution de l'hauteur des rides.

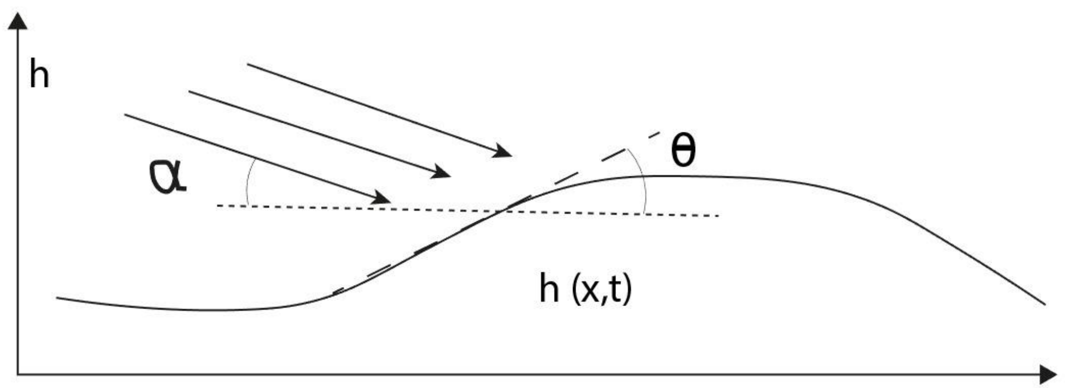


Illustration. 4 : Profil de rides et sable incident

h : Hauteur des rides

θ : Angle de la tangente

α : Angle d'incidence

Vers un modèle plus avancé 2/8

b. Modèle d'Anderson:

Hypothèse :

- Tous les grains sont similaires
- Un régime de saltation établi
- Le rebond d'un grain en saltation en éjecte 2 autres par reptation.
- Chaque grain éjecté parcourt une distance l_R et s'arrête.

- Grains en reptation: Les grains éjecté par collision avec les du lit de rides

Flux massique de grain en reptation: Q_R

- Grains en saltation: N'intervenant pas dans la composition du lit

Flux massique de grain en saltation : Q_S

Vers un modèle plus avancé 3/8

b. Modèle d'Anderson:

Par conservation de masse: $$\frac{\partial h}{\partial t} = -\frac{1}{\rho_{lit}} \frac{\partial Q}{\partial x}$$

Q : le flux total massique

ρ_{lit} : masse volumique du lit

Soit N_{ej} *le nombre de grain éjecté*

$Q(x) = Q_R(x) + Q_S(x)$, Avec :

$$\begin{cases} Q_R(x) = m_\mathrm{p} \int_{x-l_R}^{x} \mathrm{N_{ej}}(x')dx' \\ Q_S(x) = C \end{cases}$$

$Q_S(x) = C$: C est une constante de x

$$\frac{\partial Q}{\partial x} = \frac{\partial Q_R}{\partial x} \Rightarrow \frac{\partial Q}{\partial x} = m_\mathrm{p}[\mathrm{N_{ej}}(x) - \mathrm{N_{ej}}(x - l_R)]$$

$$\frac{\partial h}{\partial t} = -d^3[\mathrm{N_{ej}}(x) - \mathrm{N_{ej}}(x - l_R)] \qquad (*)$$

Vers un modèle plus avancé 4/8

c. Résolution :

Déterminons N_{ej} :

Soit $\mathrm{N_{imp}}$ le nombre de grain incident en $m^{-2}s^{-1}$
Soit $\mathrm{n_0}$ le nombre de grain éjecté par rebond

$$\mathrm{N_{ej}}(x) = \mathrm{n_0 N_{imp}}(x)$$

Pour un x fixé : $\mathrm{N_{imp}}(x)a = \mathrm{N_0} \cos\theta \left(1 + \frac{\tan\theta}{\tan\alpha}\right)$ (**)

Avec $\mathrm{N_0}$ le nombre de grain qui arrive sur une surface horizontale par unité de temps et de surface.

Nous étudions les fluctuation de la hauteur du lit par reptation on recherche alors des solution de h de la forme $e^{ikx+\omega t}$

(*) (**)

Relation de dispersion

$$\omega = ik \cot\alpha \left[1 - e^{-ikl_R}\right]$$

Avec :

$\mu_0 = N_0\, n_0\, d^3$

k : Nombre d'onde

Im(ω) : Pulsation

Re(ω): Taux de croissance

Vers un modèle plus avancé 5/8

c. Résolution :

$$\omega = ik\cot\alpha\left[1 - e^{-ikl_R}\right]$$

Partie réelle

Taux de croissance

$$Re(\omega) = \mu_0 k \cot\alpha \sin kl_R$$

DIVERGENCE

On a une divergence du taux de croissance pour les grandes valeur de k.
On a considéré précédemment que l_R constante pour tout grain or chaque grain éjecté devrait avoir une longueur de reptation propre. Anderson propose donc une distribution de la longueur de reptation.

Modélisation de la distribution des longueurs de reptation

Anderson propose de modéliser cette distribution en désignant $p(l_R)\,dl_R$ comme la probabilité que la longueur de reptation soit entre l_R et $l_R + dl_R$

$$\frac{\partial h}{\partial t} = -d^3 \int_0^{+\infty} p(l_R)\left[N_{ej}(x) - N_{ej}(x - l_R)\right] dl_R$$

Vers un modèle plus avancé 6/8

c. Résolution :

D'après des résultats expérimentaux (extrait de [1]) on peut approcher cette probabilité par une fonction.

$$p(l_R) = \frac{4}{\overline{l_R}^{2}}\, l_R\, e^{-2\frac{l_R}{\overline{l_R}}}$$

$$\omega = -\mu_0 i k \cot\alpha \left[1 - \frac{\left(1 - ik\overline{l_R}/2\right)^2}{\left(\left(1 + k^2\overline{l_R}\right)^2/4\right)^2}\right]$$

Partie réelle

Taux de croissance

$$Re(\omega) = \frac{8\,k^2 \cot\alpha}{\left(1 + k^2\,\overline{l_R}\right)^2}$$

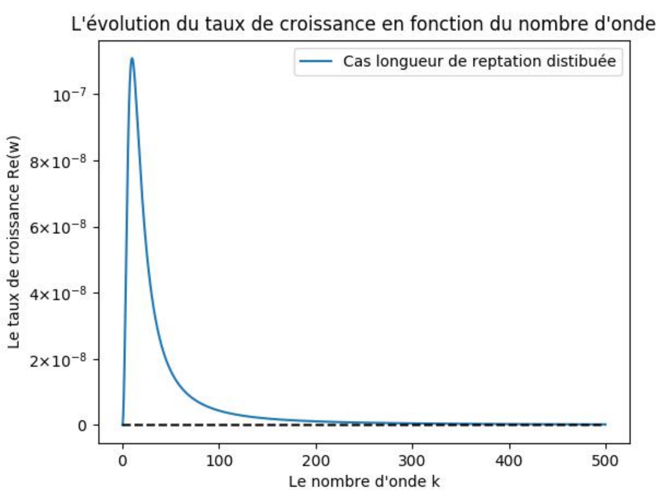


Fig. 3.1

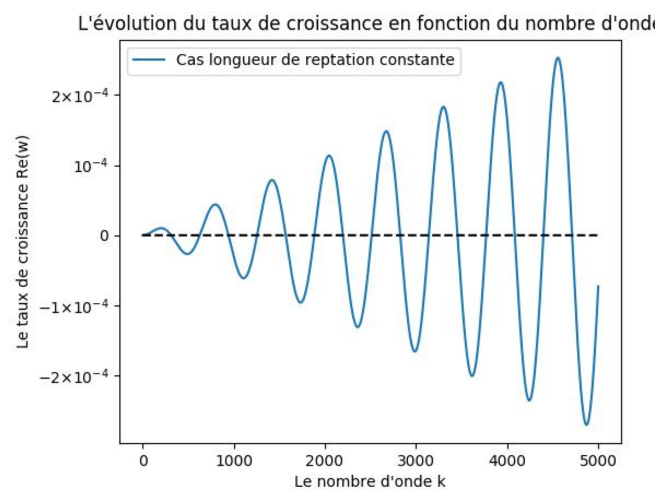


Fig. 4

Vers un modèle plus avancé 7/8

c. Résolution :

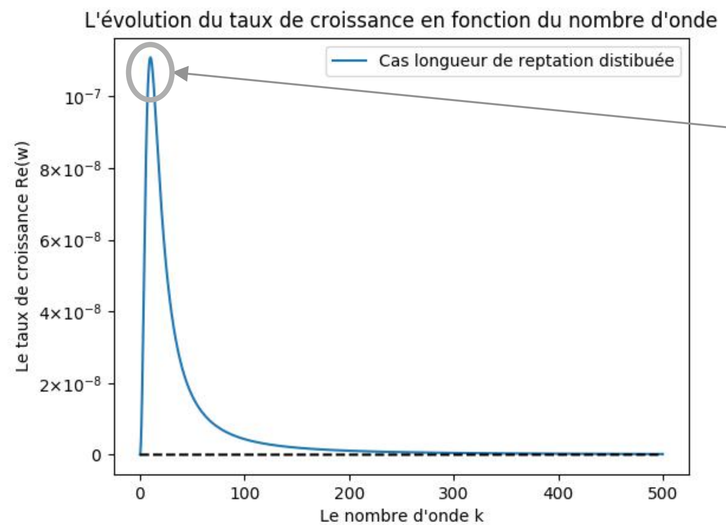


Fig. 3.1

On observe un maxima, atteint pour k_{max}, ce dernier correspond à la longueur de ride $\lambda = \frac{2\pi}{k_{max}}$ la plus instable.

Python nous donne ces valeurs:

```
Le taux de croissance est maximale pour un nombre d'onde k de l'ordre de 10.0 .
La longueur des rides devrait donc être de l'ordre de 0.628 m.
```

Fig. 3.2

Vers un modèle plus avancé 8/8

Un lit de sable composé de rides espacé de 0.628 m .

+

Un vent saturé en sable et un flux de sable incident à 14° de 100 graine par m^2 et par seconde.

=

Un fluctuation temporelle importante de la hauteur du lit .

↓

Un transport de sable s'est produit.

Conclusion

1

La saltation transporte le sable par le vent en l'éloignant d'environ 1 mètre de son point d'envol.

2

Le rebond d'un grain en saltation déclenche par collision le mouvement de grain en reptation.

3

Ces grains en reptation vont causer des fluctuations du profil du sol, des rides seront formé.

4

Pour une longueur spécifique de l'espacement des rides le transport de sable serait important.

Annexes

Fig. 1.1 1.2 1.3: **Code Python 1**

Fig. 2: Tableau extrait de *<u>Etude de quelques aspects du transport éolien des matériaux granulaires: Processus de saltation et formation des rides.</u>* **François RIOUAL**, Université Rennes 1, 2002

Fig. 3.1 et 3.2: **Code Python 2**

Fig. 4: **Code Python 3**

Animation: **Code Python 4**

Toutes les illustrations ont étaient dessiné sur Adobe Photoshop par moi-même s'inspirant d'illustrations de la thèse de François RIOUAL.

Annexes

Code Python 1

```python
import matplotlib.pyplot as plt
import math as m

#Fonction de x seconde
def f1(x,y,u):
    k1=0.6*1.184*(0.0004**2)*(6/4)
    k2=0.8*0.6*1.184*(0.0004**2)*(6/4)
    a=2500.0*(0.0004**3)
    n=((x-u)**2+y**2)**(1/2)
    f=(k1*(u-x)-k2*y)*n/a
    return f
#Fonction de y seconde
def f2(x,y,u):
    k1=0.6*1.184*(0.0004**2)*(6/4)
    k2=0.8*0.6*1.184*(0.0004**2)*(6/4)
    a=2500.0*(0.0004**3)
    n=((x-u)**2+y**2)**(1/2)
    g=(2500-1.184)*(0.0004**3)*9.81
    f=(k2*(x-u)-k1*y)*n/a-g/a
    return f

#Calcul des coefficients de la methode Runge-Kutta d'ordre 4
def kutta1(u,w,v,h):
    kx1=u*h
    ky1=w*h
    ku1=f1(u,w,v)*h
    kw1=f2(u,w,v)*h

    return kx1,ky1,ku1,kw1

def kutta23(u,w,ku1,kw1,v,h):

    kx2=(u+(ku1/2.0))*h
    ky2=(w+(kw1/2.0))*h
    ku2=f1(u+ku1/2.0,w+kw1/2.0,v)*h
    kw2=f2(u+ku1/2.0,w+kw1/2.0,v)*h

    return kx2,ky2,ku2,kw2

def kutta4(u,w,ku1,kw1,v,h):

    kx4=(u+(ku1))*h
    ky4=(w+(kw1))*h
    ku4=f1(u+ku1,w+kw1,v)*h
    kw4=f2(u+ku1,w+kw1,v)*h

    return kx4,ky4,ku4,kw4

#Resolution de l'equation differentielle
def resolution(x0,y0,u0,w0):
    h=0.001
    x=x0
    X=[x0]
    y=y0
    Y=[y0]
    u=u0
    w=w0

    for i in range(1000000):
        kx1,ky1,ku1,kw1=kutta1(u,w,5,h)
        kx2,ky2,ku2,kw2=kutta23(u,w,ku1,kw1,5,h)
        kx3,ky3,ku3,kw3=kutta23(u,w,ku2,kw2,5,h)
        kx4,ky4,ku4,kw4=kutta4(u,w,ku3,kw3,5,h)

        x += (kx1 + 2*kx2 + 2*kx3 + kx4) / 6.0
        y += (ky1 + 2*ky2 + 2*ky3 + ky4) / 6.0
        u += (ku1 + 2*ku2 + 2*ku3 + ku4) / 6.0
        w += (kw1 + 2*kw2 + 2*kw3 + kw4) / 6.0
        X.append(x)
        Y.append(y)

        if y<=0: #Condition de l'arret du programme

            return X,Y,X[i-2],Y[i-2],x,y,u,w

    return print(" Il faut plus d'itération pour que le grain atteigne le sol. ")

#Fonction permettant de calculer les conditions a implementer dans le programme pour le prochain saut
def rebond(x1,y1,x2,y2,vx,vy):

    u=x2-x1
    w=y2-y1
    theta=m.atan(w/u)
    n=0.6*((vx**2+vy**2)**(1/2))
    u,w=n*m.cos(-theta),n*m.sin(-theta)

    return u,w,theta


x2,y2,vx,vy=0,0,1,2.5
X,Y,x1,y1,x2,y2,vx,vy = resolution(x2,y2,vx,vy)

print("La vitesse d'impact est de {} m/s".format(round(((vx)**2+vy**2)**(1/2)),2))

vx,vy,theta = rebond(x1,y1,x2,y2,vx,vy)
print("L'angle d'incidence est {} degree".format(round((abs(theta)*180.0)/m.pi,2)))
```

Annexes

Suite code Python 1

```
x2,y2,vx,vy=0,0,1,2.5
X,Y,x1,y1,x2,y2,vx,vy = resolution(x2,y2,vx,vy)

print("La vitesse d'impact est de {} m/s".format(round(((vx)**2+vy**2)**(1/2)),2))

vx,vy,theta = rebond(x1,y1,x2,y2,vx,vy)
print("L'angle d'incidence est {} degree".format(round((abs(theta)*180.0)/m.pi,2)))

plt.xlim(-0.01, 1.25)
plt.ylim(0,0.15)
plt.title("Mouvement d'un grain de sable d'apres le PFD")
plt.xlabel("La distance par rapport au point d'envol (en m)")
plt.ylabel("L'altitude (en m)")
plt.plot(X,Y)
plt.show() #fig 1.1

X2,Y2,x1,y1,x2,y2,vx,vy=resolution(x2,0,vx,vy)
vx,vy,theta = rebond(x1,y1,x2,y2,vx,vy)

plt.xlim(-0.01, 1.25)
plt.ylim(0,0.15)
plt.title("Mouvement d'un grain de sable d'apres le PFD")
plt.xlabel("La distance par rapport au point d'envol (en m)")
plt.ylabel("L'altitude (en m)")
plt.plot(X,Y)
plt.plot(X2,Y2)
plt.show() #fig. 1.2


X3,Y3,x1,y1,x2,y2,vx,vy=resolution(x2,0,vx,vy)
vx,vy,theta = rebond(x1,y1,x2,y2,vx,vy)
X4,Y4,x1,y1,x2,y2,vx,vy=resolution(x2,0,vx,vy)

plt.xlim(-0.01, 1.25)
plt.ylim(0,0.15)
plt.title("Mouvement d'un grain de sable d'apres le PFD")
plt.xlabel("La distance par rapport au point d'envol (en m)")
plt.ylabel("L'altitude (en m)")
plt.plot(X,Y)
plt.plot(X2,Y2)
plt.plot(X3,Y3)
plt.plot(X4,Y4)
plt.show() #fig 1.3
```

Annexes

Code Python 2

```
import matplotlib.ticker as mticker
import matplotlib.pyplot as plt
import math as m

a=(1/m.tan(0.2268928))*(0.0004**3)*2.0*100

X=[0.0001*i for i in range(5000000)]
Y2=[8*a*0.01*(x**2)*((1+0.01*(x**2))**(-2)) for x in X]
Y3=[0 for x in X]
M = max(Y2)
i = Y2.index(M)
kmax=round(X[i],3)
lambd=round((2*m.pi)/kmax,3)
print("Le taux de croissance est maximale pour un nombre d'onde k de l'ordre de",kmax,".")
print("La longueur des rides devrait donc être de l'ordre de",lambd,"m.")
plt.plot(X,Y2,label='Cas longueur de reptation distibuée')
plt.plot(X,Y3, color='black', linestyle='dashed')
plt.legend()
f = mticker.ScalarFormatter(useOffset=False, useMathText=True)
g = lambda x,pos : "${}$".format(f._formatSciNotation('%1.10e' % x))
plt.gca().yaxis.set_major_formatter(mticker.FuncFormatter(g))
plt.xlabel("Le nombre d'onde k")
plt.ylabel("Le taux de croissance Re(w)")
plt.title("L'évolution du taux de croissance en fonction du nombre d'onde")
plt.show()
```

Annexes

Code Python 3

```python
import matplotlib.ticker as mticker
import matplotlib.pyplot as plt
import math as m
a=(1/m.tan(0.2268928))*(0.0004**3)*2.0*100

X=[0.01*i for i in range(500000)]
Y1=[m.sin(x*0.01)*x*a for x in X]
Y3=[0 for x in X]
plt.plot(X,Y1, label='Cas longueur de reptation constante')
plt.legend()
plt.plot(X,Y3, color='black', linestyle='dashed')
f = mticker.ScalarFormatter(useOffset=False, useMathText=True)
g = lambda x,pos : "${}$".format(f._formatSciNotation('%1.10e' % x))
plt.gca().yaxis.set_major_formatter(mticker.FuncFormatter(g))
plt.xlabel("Le nombre d'onde k")
plt.ylabel("Le taux de croissance Re(w)")
plt.title("L'évolution du taux de croissance en fonction du nombre d'onde")
plt.show()
```

Annexes

Code Python 4

```python
def animation(P,H,h):
    t=0
    X=[]
    Y=[]
    for i in range(len(P)):
        t+=h
        x=P[i]
        y=H[i]
        X.append(x)
        Y.append(y)
        temps='Temps: {} s'.format(round(t,4))
        text=ax.text(0.8, 0.1, temps , fontsize=10)
        point,=plt.plot(x, y, color='black', marker='.') #Le grain de sable
        line,=plt.plot(X,Y, color='blue',linestyle='dashed') #La trajectoire
        plt.pause(h) #Pause pour l'animation
        point.remove()
        text.remove()
        line.remove()

    return

x2,y2,vx,vy=0,0,1,2.5
X1,Y1,x1,y1,x2,y2,vx,vy = resolution(x2,y2,vx,vy)
vx,vy,theta = rebond(x1,y1,x2,y2,vx,vy)
X2,Y2,x1,y1,x2,y2,vx,vy=resolution(x2,0,vx,vy)
vx,vy,theta = rebond(x1,y1,x2,y2,vx,vy)
X3,Y3,x1,y1,x2,y2,vx,vy=resolution(x2,0,vx,vy)
vx,vy,theta = rebond(x1,y1,x2,y2,vx,vy)
X4,Y4,x1,y1,x2,y2,vx,vy=resolution(x2,0,vx,vy)


fig = plt.figure()
ax = fig.add_subplot()
plt.xlim(-0.01, 1.25)
plt.ylim(0,0.15)
plt.title("Mouvement d'un grain de sable d'apres le PFD")
plt.xlabel("La distance par rapport au point d'envol (en m)")
plt.ylabel("L'altitude (en m)")

X,Y = X1+X2+X3+X4 , Y1+Y2+Y3+Y4

animation(X,Y, 0.001)
```

Annexes