

Travaux de Modélisation et de Simulation Déplacement d'une dune sous l'action du vent

Mathieu Faurie - François Strina

26 mars 2004

ABSTRACT

We present in this paper a study of the evolution of a sand dune in a air flow. This problem belongs to the class of the problems which join together a granular cluster and a fluid flow. The difficulty for resolving such problems comes from the non-linearity of the Navier-Stokes equations and the interactions that exist between the fluid and the granular cluster. We will exhibit the results based on a very simplified model and computed with *CASTEM*.



Table des matières

Introduction	3
1 Analyse du problème	4
1.1 Hypothèses	4
1.2 Modèle utilisé	5
1.3 Calcul du frottement pariétal	6
1.3.1 Cas du fluide parfait : solution linéarisée	6
1.3.2 Cas du fluide réel	7
2 Etude statique	8
2.1 Code de calcul	8
2.2 Choix des paramètres	8
2.2.1 Taille du domaine	8
2.2.2 Finesse du maillage	9
2.2.3 Critère de convergence	9
2.3 Résultats obtenus	10
2.3.1 Influence de ϵ_0	10
2.3.2 Influence du nombre de Reynolds	12
2.3.3 Influence de l'intensité du cisaillement	12
2.3.4 Influence de la hauteur de la dune	14
2.3.5 Conclusion	15
3 Application au calcul de l'évolution d'un « système dunaire »	16
3.1 Structure du programme	16
3.2 Etude de l'influence des différents paramètres	17
3.2.1 Profil initial de la dune	17
3.2.2 Paramètre τ_S	19
3.2.3 Paramètre Λ	20
3.3 Relachement des hypothèses de départ	21
3.4 Résultats obtenus dans le cas linéaire	21
Conclusion	23
Annexe A : Programme C	24
Annexe B : Code <i>CASTEM</i>	27
Références	29

Introduction

En dépit des nombreuses recherches réalisées ces dernières années, la question de l'évolution des dunes de sable reste un phénomène mal compris. Ceci peut sembler paradoxal si l'on songe au fait qu'il s'agit là d'un phénomène ancien et universel : il existe en effet de nombreuses variantes à ce problème, toute combinaison d'un fluide et d'un amas granulaire appartenant à la même classe de problèmes que celui qui nous intéresse ici. Les difficultés principales associées à ce type de problème résident dans la non-linéarité des équations régissant le fluide (équations de Navier-Stokes) ainsi que dans les interactions qui existent entre le fluide et l'amas granulaire. On peut néanmoins penser qu'avec les progrès réalisés dans les méthodes de résolution numérique et la puissance sans cesse croissante des moyens de calculs, la compréhension de ces phénomènes progressera rapidement au cours des prochaines années.

Ce rapport a pour objectif une meilleure compréhension des phénomènes de base sous-jacents à l'évolution d'une dune. Dans une première partie, nous définirons notre cadre d'étude en précisant nos hypothèses et le modèle qui en découle. Nous présenterons ensuite brièvement notre façon de procéder. La deuxième partie sera consacrée à l'étude statique : validation du code de calcul de l'écoulement utilisé et du choix des paramètres de simulation puis analyse des résultats. Enfin, dans la troisième partie, nous réaliserons le passage du modèle statique au modèle dynamique en nous appuyant sur les observations de la partie précédente. Les résultats dynamiques seront interprétés et confrontés à ceux obtenus à l'aide d'un programme résolvant le problème linéarisé (cas du fluide parfait).

1 Analyse du problème

La question de l'évolution d'une dune sous l'action du vent est un problème de type fluide-structure tel que

$$\frac{T_{Fluide}}{T_{Solide}} \ll 1,$$

où T_{Fluide} est l'ordre de grandeur du temps mit par le fluide (l'air dans le cas présent) à parcourir la dune, et T_{Solide} un temps caractéristique de la dynamique de la dune. En effet, T_{Fluide} est de l'ordre de la seconde alors que T_{Solide} vaut typiquement plusieurs heures.

1.1 Hypothèses

Nous sommes donc ici dans le cadre des hypothèses de l'aéroélasticité quasi-statique :

- à l'échelle de temps du fluide, le solide semble immobile, ce dernier étant seulement déformé par les efforts résultant de l'écoulement ;
- du point de vue du solide, le fluide n'est vu que comme une source d'efforts statiques.

Il est donc parfaitement légitime de découpler le traitement des dynamiques du fluide et du solide et d'utiliser les résultats de calculs avec le solide immobile pour estimer le mouvement de ce solide (Figure 1).

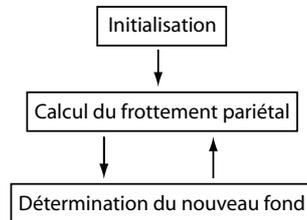


FIG. 1 – Calcul dans l'hypothèse d'aéroélasticité

Afin de se ramener à un problème susceptible d'être traité dans le cadre des TMS, nous avons été contraints d'adopter plusieurs hypothèses simplificatrices :

- le sol sur lequel repose la dune est considéré comme dur ; aucun « creusement » ne pourra donc être observé,
- nous nous intéressons à l'évolution d'une dune unique supposée bidimensionnelle (2D) transverse (invariante dans la direction perpendiculaire au vent),
- la dune est supposée de hauteur suffisamment faible pour appartenir en totalité à la zone linéaire de la couche limite laminaire¹ (Figure 2).

¹la solution de Blasius pouvant être interpolée linéairement dans la partie inférieure de la couche limite

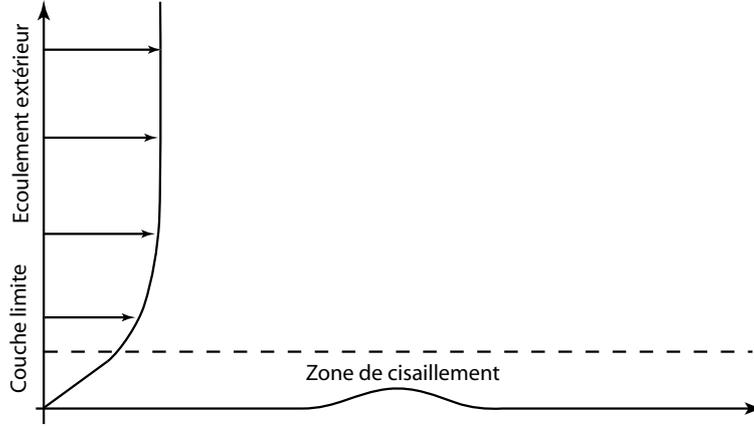


FIG. 2 – Structure d’une couche limite laminaire

1.2 Modèle utilisé

Pour permettre à la dune d’évoluer au cours du temps, nous utilisons deux lois :

- une traduisant la conservation de la masse, reliant la variation temporelle de la forme du fond à la divergence du flux de matière :

$$\frac{\partial f}{\partial t} = -\frac{\partial q}{\partial x} \quad (1)$$

- une reliant le flux de matière à la valeur du frottement pariétal :

$$q = \varpi(\tau - \tau_s - \Lambda \frac{\partial f}{\partial x}) \quad \text{où } \varpi(x) = \begin{cases} 0 & \text{si } x < 0 \\ x & \text{si } x \geq 0 \end{cases} \quad (2)$$

L’équation portant sur q traduit le fait que plus le frottement pariétal τ est élevé plus la matière est emporté par l’écoulement [1]. τ_s traduit l’existence d’un seuil pour l’arrachement : la dune est plus ou moins meuble. Le terme $\Lambda \frac{\partial f}{\partial x}$ permet de tenir compte de la gravité : le seuil est modifié avec le signe de la pente, une contrainte supérieure étant nécessaire pour une pente positive.

En réalité, nous n’utilisons pas l’équation (2), mais l’équation modifiée suivante :

$$\frac{\partial q}{\partial x} = \frac{q_{sat} - q}{l_{sat}} \quad , \quad q_{sat} = \varpi(\tau - \tau_s - \Lambda \frac{\partial f}{\partial x}) \quad (3)$$

Cette modification permet « d’adoucir » les variations de q et de lui faire atteindre sa valeur maximale q_{sat} au bout d’une distance caractéristique l_{sat} . En effet, on constate expérimentalement que lorsque le vent souffle sur le sable le flux de matière varie comme représenté Figure 3. On consultera [2] pour plus de précisions sur ce sujet.

Pour faire évoluer la dune nous avons donc besoin de connaître la forme du sol f et la valeur du frottement pariétal τ . Nous allons maintenant nous pencher sur deux méthodes différentes permettant d’obtenir le frottement pariétal.

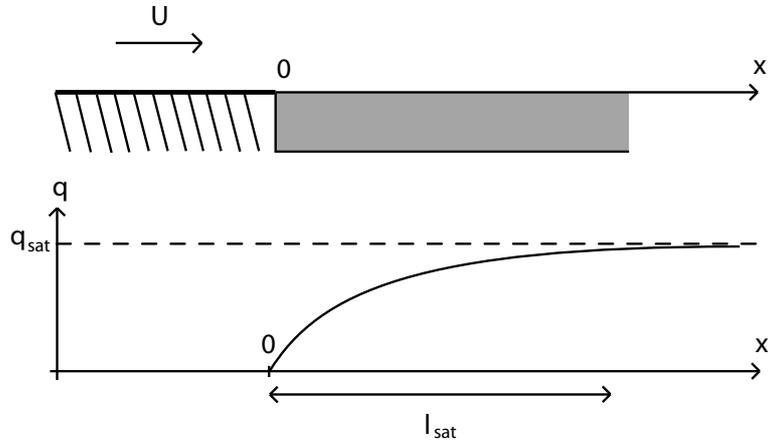


FIG. 3 – Variations de q

1.3 Calcul du frottement pariétal

1.3.1 Cas du fluide parfait : solution linéarisée

On considère le fluide comme parfait ($Re = \infty$) et on cherche à calculer le frottement pariétal sur une dune de petite hauteur ($h_0 \ll 1$) dans le cas d'un cisaillement (Figure 4). On peut alors montrer que le frottement pariétal τ est donné par la formule suivante (voir [4] pour la démonstration), où FT désigne la transformée de Fourier et Ai la fonction d'Airy :

$$FT[\tau] = 3Ai(0)(-ik)^{\frac{1}{3}}FT[f] \quad (4)$$

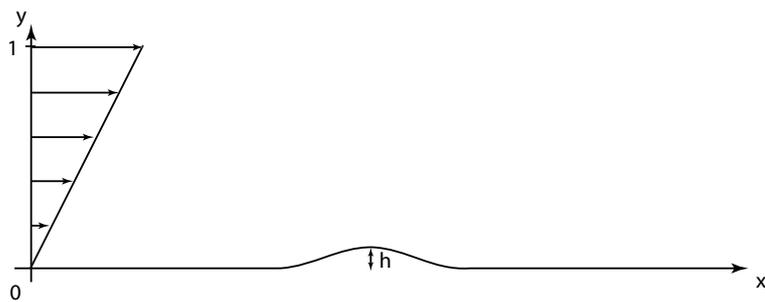


FIG. 4 – Problème étudié

Trouver τ revient donc à calculer des transformées de Fourier, ce qui est assez rapide et permet d'obtenir facilement l'évolution de la dune au cours du temps. Un programme fourni par notre tuteur, M. P-Y. Lagrée, nous a permis de résoudre cette étape de calcul et de faire ainsi évoluer la dune au cours du temps.

1.3.2 Cas du fluide réel

Lorsque le fluide n'est plus considéré comme parfait, la formule précédente ne s'applique plus. Il est alors nécessaire de résoudre à chaque pas de temps l'équation de Navier-Stokes pour le problème étudié. Nous avons utilisé *CASTEM 2001*, code de calcul développé par le CEA, afin de réaliser cette étape. Nous allons donc étudier dans ce qui suit les résultats obtenus avec *CASTEM* afin de déterminer les caractéristiques de l'écoulement et de choisir au mieux les paramètres du problème.

2 Etude statique

2.1 Code de calcul

Afin de calculer l'écoulement autour de la dune, nous avons utilisé le module de résolution des équations de Navier-Stokes de *CASTEM 2001*. Si sa polyvalence n'est plus à démontrer, en revanche son ergonomie et sa convivialité sont plus discutables. La phase de prise en main du code, phase préliminaire à tout calcul proprement dit, s'en est donc trouvé rallongée d'autant. Une fois les principes de base assimilés, nous avons pu débiter les calculs d'écoulement autour d'une dune. Pour ce faire, nous sommes partis d'un fichier qui nous a été fourni par P.-Y. Lagrée, fichier que nous avons ensuite modifié pour l'adapter à nos besoins.

Pour calculer l'écoulement, le code discrétise les termes de diffusion et de convection et procède par itérations successives. Toutes les grandeurs utilisées sont adimensionnées.

2.2 Choix des paramètres

Le choix de certains paramètres revêt une importance capitale dans la mesure où la validité des solutions obtenues en dépend fortement. Les principaux paramètres qui nous préoccuperont ici sont la taille du domaine, la finesse de son maillage et le critère de convergence. Avant toute chose, on peut souligner que le choix de ces paramètres est une question de compromis : une augmentation de la taille du domaine, un affinement du maillage ou un critère de convergence plus sévère concourent chacun à l'amélioration de la solution trouvée ; cependant, chacune de ces modifications augmente les temps de calcul. Or il faut garder à l'esprit que notre but est de calculer l'évolution d'une dune au cours du temps. Il est par conséquent indispensable de dimensionner correctement ces paramètres afin de se limiter à des temps de calcul de l'ordre de quelques minutes par pas de temps.

2.2.1 Taille du domaine

A l'évidence, la présence de la dune va modifier l'écoulement dans son voisinage. Or un domaine correctement dimensionné est un domaine qui prend en compte l'ensemble du phénomène. Notre domaine doit donc être tel que cette « zone de perturbation » soit entièrement incluse dans celui-ci : aux frontières du domaine, l'écoulement doit être quasiment le même que celui qui existerait en l'absence de toute obstacle. Par conséquent, il paraît naturel de penser que la partie du domaine en amont de la dune peut être plus petite que celle en aval.

2.2.2 Finesse du maillage

La première remarque qui peut être faite est que l'essentiel des phénomènes qui se produisent se situent dans le voisinage immédiat de la dune et du sol en général. Il semble donc logique d'affiner le maillage dans cette zone (y petit).

Dans le même ordre d'idée, nous avons dans un premier temps pensé à préciser le maillage dans une « tranche » verticale contenant la dune (abscisses voisines de l'abscisse de hauteur maximale $x_{h_{max}}$). Si ceci était astucieux et facile à mettre en oeuvre dans le cas d'une dune statique (un pas de temps), c'était beaucoup plus délicat à implémenter pour une dune mobile. Le maillage est donc régulier suivant l'axe du vent.

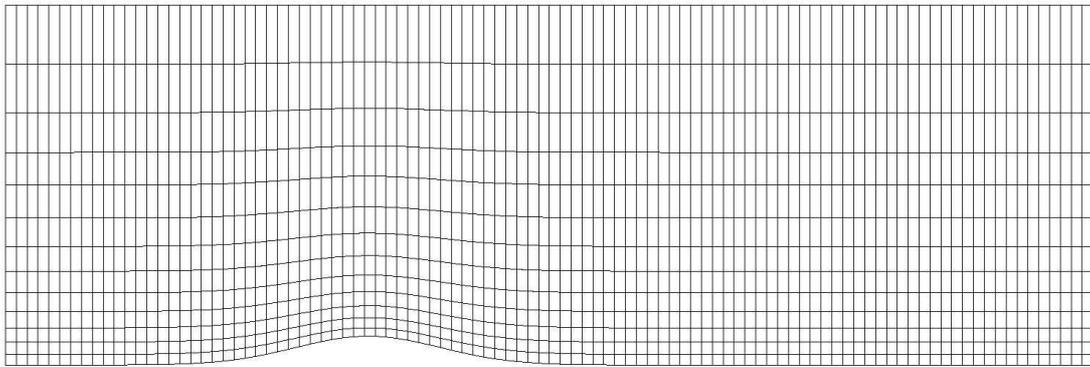


FIG. 5 – Exemple de maillage utilisé

2.2.3 Critère de convergence

Comme expliqué précédemment, pour calculer le champ de vitesse *CASTEM* procède par itérations successives. Sachant ceci, il faut se demander à quel moment la solution obtenue peut être jugée satisfaisante et donc quand il est légitime d'arrêter le calcul.

Pour juger de « l'état de convergence » d'une solution, nous avons donc introduit un calcul d'erreur entre deux itérations successives. Ainsi, à chaque itération nous calculons, pour chacun des noeuds du maillage, les valeurs des erreurs

$$\epsilon_U = |U_n - U_{n-1}| \text{ et } \epsilon_V = |V_n - V_{n-1}|$$

où U et V sont les composantes du champ de vitesse et n et $(n - 1)$ les indices caractérisant respectivement l'itération actuelle et l'itération précédente.

Un test implémenté dans le fichier utilisé par *CASTEM* interrompt le calcul lorsque la condition suivante est vérifiée :

$$Max_{Maillage}(\epsilon_U, \epsilon_V) < \epsilon_0.$$

où ϵ_0 est l'erreur à partir de laquelle nous considérons que la solution obtenue peut être qualifiée de *convergée*.

2.3 Résultats obtenus

Afin de déterminer l'influence de certains paramètres (critère de convergence ϵ_0 , nombre de Reynolds Re , intensité du cisaillement U , hauteur de la dune h) sur les résultats obtenus, nous avons réalisé de nombreuses simulations pour différentes valeurs mais en conservant la même dune et le même domaine (domaine rectangulaire de hauteur 1 et de largeur 3, x variant de -1.5 à 1.5). La dune est une gaussienne centrée en $x = -0.5$ de hauteur $h_0 = 0.10$ et de demi-largeur 0.5 (Figure 6). Nous avons choisi de décentrer la dune afin de mieux pouvoir observer les perturbations introduites par la dune sur l'écoulement aval (l'écoulement amont nous a semblé peu perturbé).

Pour évaluer ces influences, on s'intéresse ici à l'évolution du coefficient de frottement τ et de la pression p sur le sol.

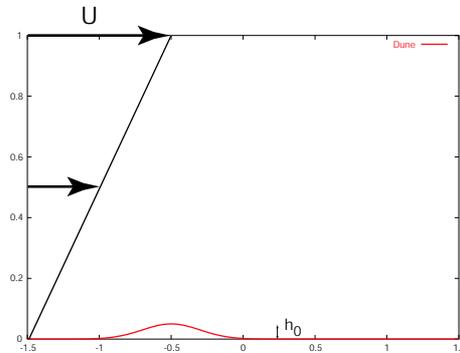


FIG. 6 – Dune étudiée

2.3.1 Influence de ϵ_0

On considère ici une dune de hauteur $h = 0.10$ dans un cisaillement d'intensité $U = 1$ pour $Re = 1000$. Le paramètre ϵ_0 prend quant à lui les valeurs 10^{-4} , 10^{-5} , et 10^{-6} . Les résultats sont présentés Figures 7 et 8.

On observe que, aussi bien pour la pression que pour le coefficient de frottement et quelque soit la valeur attribuée à ϵ_0 , les trois courbes se superposent correctement

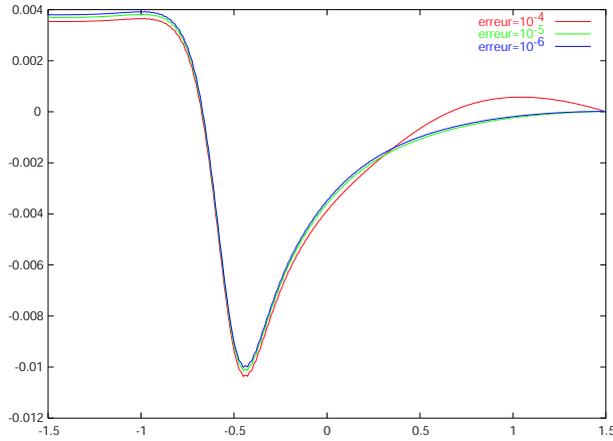


FIG. 7 – Influence de ϵ_0 : Pression sur le sol

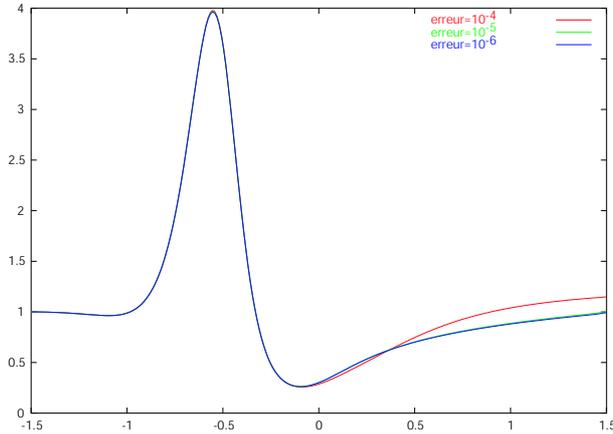


FIG. 8 – Influence de ϵ_0 : Coefficient de frottement sur le sol

avant et sur la dune. En revanche, en aval de la dune, la courbe correspondant à $\epsilon_0 = 10^{-4}$ se distingue des deux autres.

On peut donc penser qu'une valeur de 10^{-5} pour l'erreur est un bon compromis puisqu'elle permet d'obtenir une bonne précision pour un temps de calcul raisonnable (on constate en effet qu'en passant de 10^{-5} à 10^{-6} pour l'erreur, on n'améliore que très peu la précision alors que le temps de calcul est quasiment multiplié par 3).

Au vu des résultats précédents, nous avons donc pris la décision d'attribuer la valeur 10^{-5} à ϵ_0 pour la suite de l'étude en statique (dune immobile) afin de pouvoir établir correctement l'influence des autres paramètres.

2.3.2 Influence du nombre de Reynolds

On considère ici une dune de hauteur $h = 0.10$ dans un cisaillement d'intensité $U = 1$, le paramètre Re prenant les valeurs 500, 1000, et 1500. Les résultats sont présentés Figures 9 et 10.

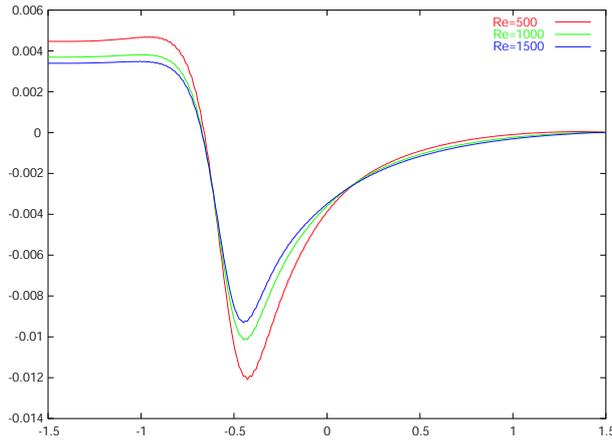


FIG. 9 – Influence du Reynolds : Pression sur le sol

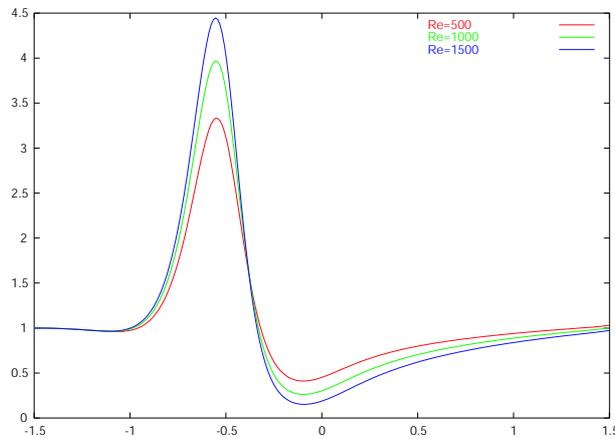


FIG. 10 – Influence du Reynolds : Coefficient de frottement sur le sol

On peut constater que, aussi bien pour la pression que le coefficient de frottement, les 3 courbes sont homothétiques : à un coefficient d'échelle près, elles sont superposables.

2.3.3 Influence de l'intensité du cisaillement

On considère ici une dune de hauteur 0.10 dans un écoulement de $Re = 1000$. L'intensité du cisaillement prend successivement les valeurs 1, 2 et 4. Les résultats

sont présentés Figures 11 et 12.

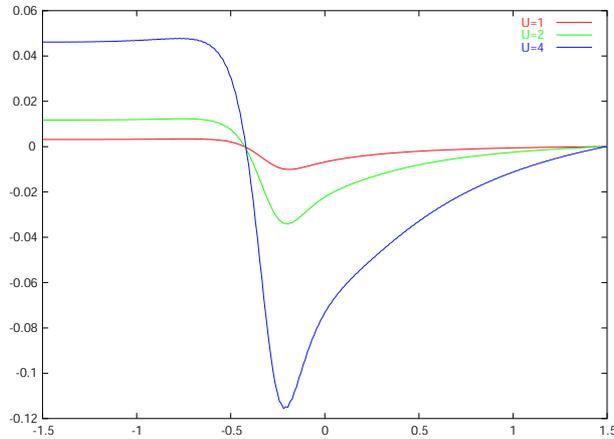


FIG. 11 – Influence de l'intensité du cisaillement : Pression sur le sol

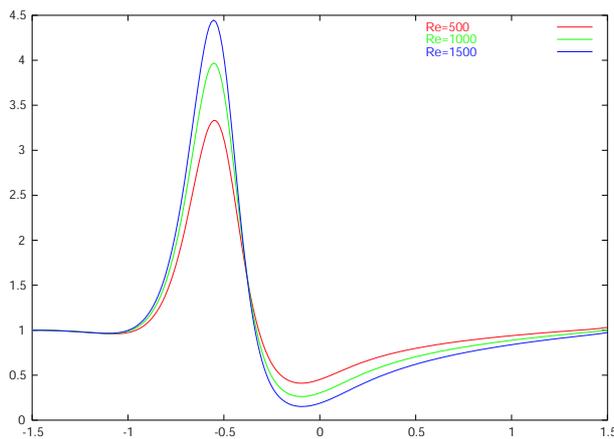


FIG. 12 – Influence de l'intensité du cisaillement : Coefficient de frottement sur le sol

L'examen des courbes concernant la pression montre que celles-ci aussi sont homothétiques. On peut noter que les valeurs de la pression semblent varier en U^2 (ce qui peut être aisément constaté en $x = 0$). Ceci n'est a priori pas choquant si l'on se réfère à la loi de Bernoulli.

Concernant le coefficient de frottement, les 3 courbes présentent globalement la même forme, à l'exception de la zone aval pour la courbe correspondant à $U = 4$. Ceci s'explique par l'existence d'une zone de recirculation derrière la bosse lorsque le cisaillement est suffisamment fort ; sa présence est confirmée par les valeurs négatives du coefficient de frottement observées juste derrière la dune. En amont de la dune,

l'écoulement est peu perturbé : la valeur est constante et prend la valeur du gradient de vitesse $\frac{\partial u}{\partial y} = U$.

2.3.4 Influence de la hauteur de la dune

On considère ici une dune dans un écoulement cisailé d'intensité $U = 1$ et de $Re = 1000$. La hauteur de la dune est variable et vaut successivement 0.05, 0.10 et 0.15. Les résultats sont présentés Figures 13 et 14.

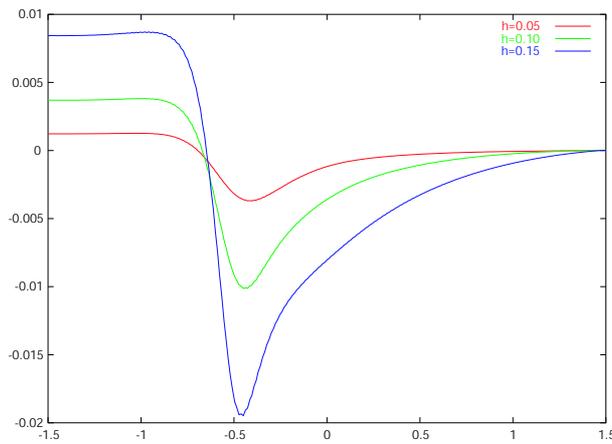


FIG. 13 – Influence de la hauteur de la dune : Pression sur le sol

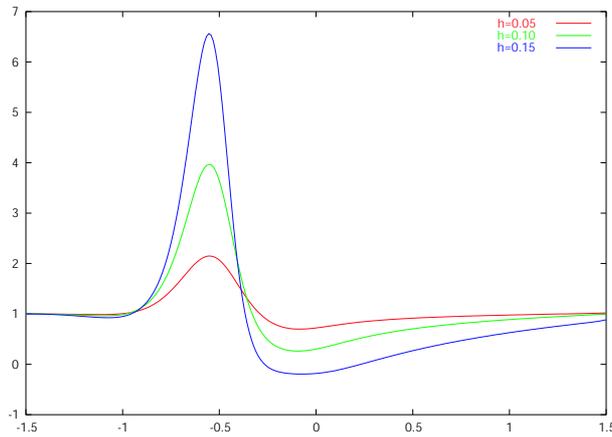


FIG. 14 – Influence de la hauteur de la dune : Coefficient de frottement sur le sol

Comme précédemment, les courbes de pression sont homogothétiques. Les courbes de coefficient de frottement, quant à elles, ont des allures similaires mais pour $h = 0.15$ on peut à nouveau constater la présence d'une zone de recirculation en aval de la dune : une dune de hauteur élevée perturbe de manière importante l'écoulement.

2.3.5 Conclusion

Quelles que soient les valeurs adoptées pour les paramètres, on observe que le minimum de pression se situe toujours un peu en aval de la dune alors que le pic de coefficient de frottement est un peu en amont.

Les résultats précédents ont montré que Castem donne, pour l'écoulement, des solutions apparemment cohérentes. Etant donné le caractère homothétique des résultats obtenus, on peut choisir n'importe quelle valeur pour chacun des paramètres sachant toutefois que si on choisit une hauteur ou un cisaillement trop importants on doit s'attendre à observer des zones de recirculation. On choisit donc pour la suite :

$$Re = 1000$$

$$U = 1$$

$$h = 0.10$$

3 Application au calcul de l'évolution d'un « système dunaire »

La solution calculée par *CASTEM* pour une dune immobile ayant été validée, nous pouvons maintenant nous intéresser à la dynamique de la dune en simulant son évolution.

3.1 Structure du programme

Nous avons écrit, en nous inspirant d'un programme fourni par M. P.-Y Lagrée, un programme en C qui résout le système constitué des équations (1) et (3) et qui est capable de « dialoguer » avec *CASTEM*. La structure algorithmique de l'ensemble est la suivante :

- les paramètres de calcul sont lus par le programme en C qui en déduit une forme de dune initiale
- *CASTEM* résout l'écoulement en tenant compte de la forme de la dune fourni par le programme en C
- l'écoulement étant calculé, *CASTEM* sauvegarde ses caractéristiques (champ de vitesse, pression sur le sol, frottement pariétal) dans un fichier
- le programme en C reprend la main et calcule une nouvelle forme de dune à partir de la forme précédente et du frottement fourni par *CASTEM*
- le processus recommence avec une dune légèrement modifiée.

Le schéma Figure 15 représente ce mode de fonctionnement.

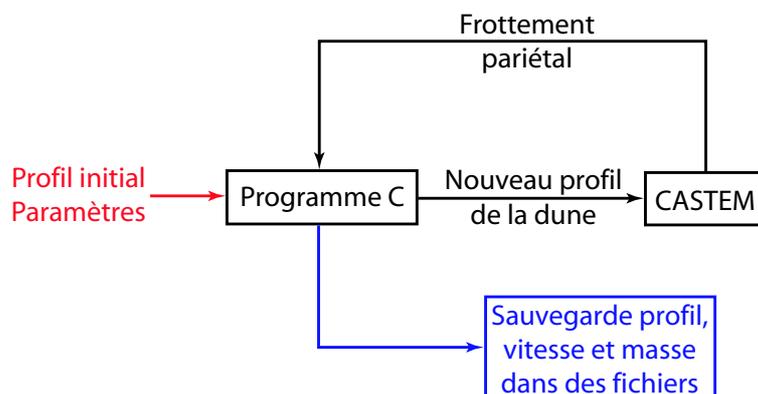


FIG. 15 – Structure du programme

3.2 Etude de l'influence des différents paramètres

Nous avons effectué différents calculs en modifiant à chaque fois un paramètre afin d'en déterminer l'influence sur les solutions obtenues.

Les objectifs de cette étude étaient différents suivant les paramètres considérés :

- pour les coefficients des équations (τ_S ou Λ par exemple), inconnus à priori, il s'agissait d'en déterminer une valeur (ou une plage de valeurs) qui donne des résultats acceptables ;
- pour les paramètres de calcul (taille du pas de temps, durée de la période d'évolution calculée, précision du maillage...), les premiers calculs étaient destinés à nous renseigner sur la valeur à donner à ces paramètres afin d'optimiser les calculs suivants en réduisant le temps de calcul mais en conservant le phénomène étudié dans le domaine d'étude ;
- en ce qui les paramètres propres à l'écoulement (profil initial de la dune, sol dur ou meuble...), il nous est apparu intéressant d'étudier leur influence sur l'évolution afin d'essayer d'en tirer des conclusions exploitables.

Après avoir réalisé quelques calculs avec un critère de convergence ϵ_0 valant 10^{-5} nous avons constaté que le temps nécessaire pour compléter un pas de temps n'était pas compatible avec le temps dont nous disposions. Nous avons donc choisi d'imposer $\epsilon_0 = 10^{-4}$, la partie 2 ayant montré que cette précision suffisait pour obtenir des résultats acceptables sur une grande partie du domaine. Nous avons pu noter qu'en combinant cette condition de convergence avec les conditions portant sur la taille du domaine et la précision du maillage, les calculs, effectués sur les ordinateurs des salles informatiques de l'Ecole Polytechnique donnaient des résultats en des temps raisonnables : typiquement, un pas de temps prend une dizaine de minutes.

Les courbes présentées dans cette partie représentent l'évolution temporelle de la dune. Elles sont construites par superposition des profils dunaires aux différents pas de temps.

3.2.1 Profil initial de la dune

Pour toutes nos simulations, nous nous sommes limités à deux formes de dunes différentes :

- soit une forme en cosinus

$$f(x) = \begin{cases} h_0(1 + \cos(2\pi \frac{x - x_0}{L_b})) & \text{si } |x - x_0| < \frac{L_b}{2} \\ 0 & \text{sinon} \end{cases}$$

- soit une forme en gaussienne

$$f(x) = h_0 \exp(-\pi \frac{(x - x_0)^2}{L_b^2})$$

avec h_0 hauteur de la dune, L_b sa demi-largeur et x_0 l'abscisse de son sommet.

Les Figures 16 et 17 représentent les résultats obtenus en ne changeant que la forme initiale de la bosse mais pas la hauteur (pour améliorer la lisibilité, les courbes ont subi une dilatation verticale).

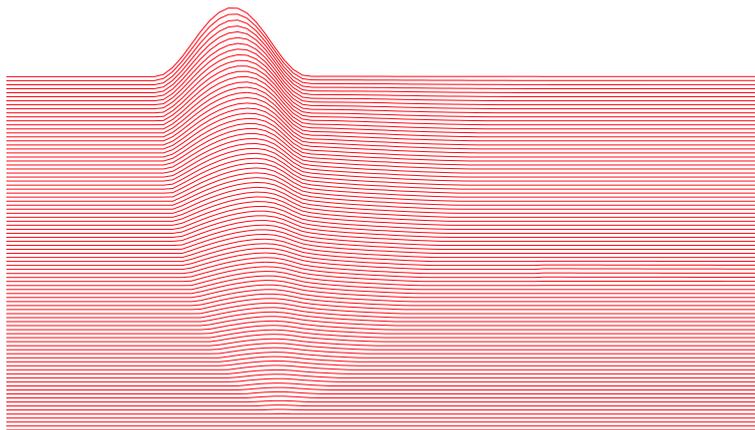


FIG. 16 – Forme initiale : cosinus

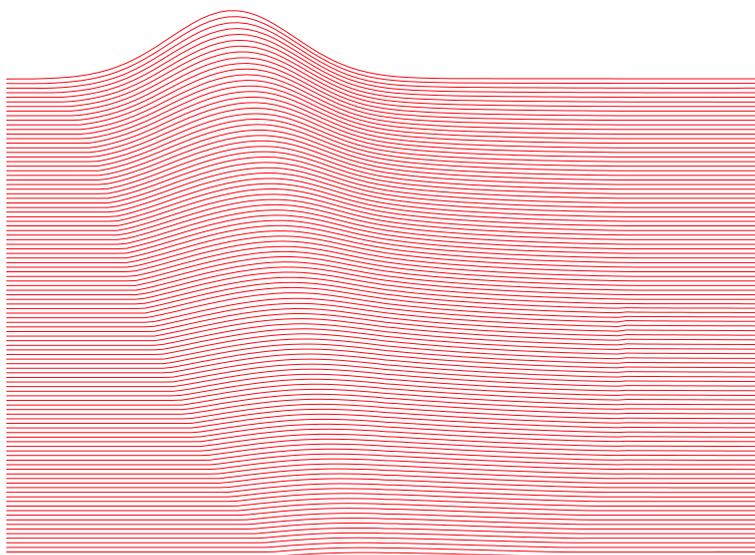


FIG. 17 – Forme initiale : gaussienne

La lecture des courbes fait apparaître une vitesse plus importante pour l'évolution de la dune dans le cas de la gaussienne. On peut en outre constater un affaissement plus rapide dans le cas du cosinus.

Ainsi, si dans les deux cas on obtient des solutions intéressantes, nous avons pris la décision de n'utiliser, pour la suite de nos simulations, que des dunes à profil gaussien pour deux raisons :

- l'affaissement étant moindre, on pourra observer une évolution sur une plage temporelle plus importante,
- la vitesse est plus importante et donc les phénomènes d'évolution seront plus aisément observables.

3.2.2 Paramètre τ_S

Les Figures 18 et 19 représentent les résultats obtenus en ne changeant que la valeur du coefficient τ_S .

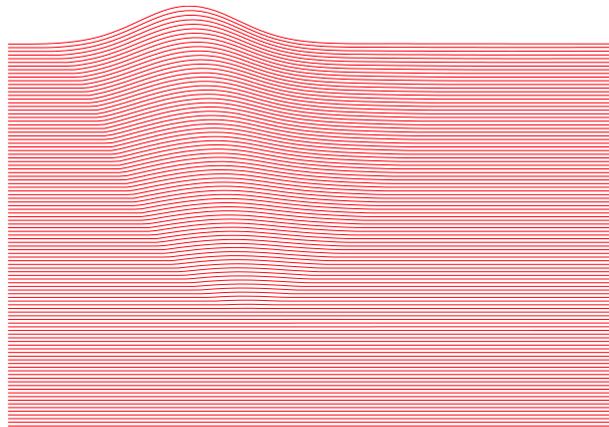


FIG. 18 – Evolution d'une dune pour $\tau_S = 0.2$

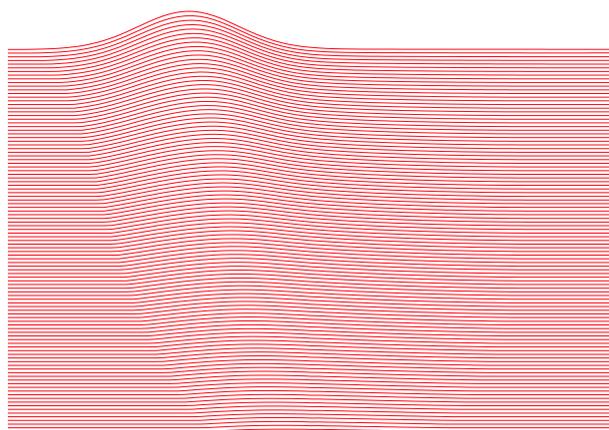


FIG. 19 – Evolution d'une dune pour $\tau_S = 0.6$

Sur ces courbes, on constate qu'une valeur de τ_S élevée se traduit par un étalement de la dune et un affaissement relativement lent. Ceci est cohérent car il est plus difficile d'« arracher » des grains de sable lorsque τ_S est élevé. En outre, la vitesse de déplacement de la dune semble être la même dans les deux cas.

3.2.3 Paramètre Λ

Les Figures 20 et 21 représentent les résultats obtenus en ne changeant que la valeur du coefficient Λ .

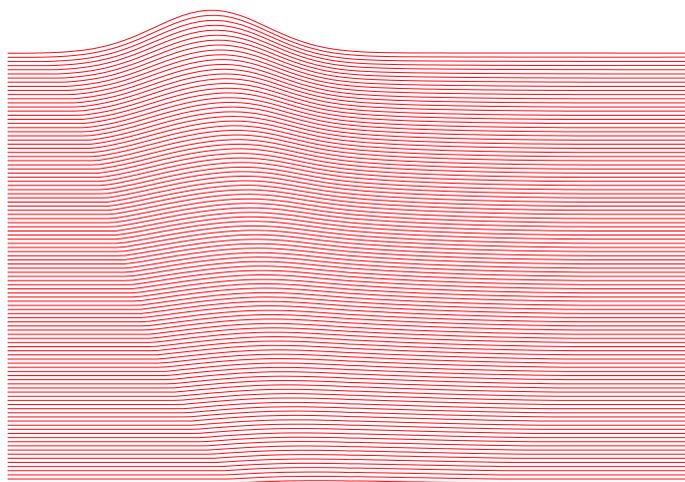


FIG. 20 – Evolution d'une dune pour $\Lambda = 0.5$

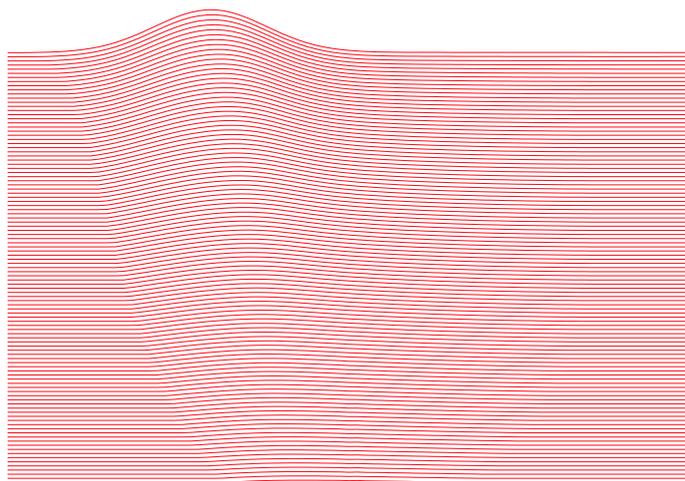


FIG. 21 – Evolution d'une dune pour $\Lambda = 1$

La lecture de ces courbes fait apparaître une vitesse de déplacement plus élevée pour un Λ faible. Ceci peut s'interpréter par le fait que le sable est prélevé sur la pente amont de la dune pour être emmené sur sa pente aval. Or Λ modélise un effet de gravité donc plus il est élevé, plus le frottement pariétal doit être important pour faire « remonter » la pente au sable.

Nous avons de plus remarqué qu'une valeur trop importante de Λ fait rapidement apparaître des instabilités.

3.3 Relachement des hypothèses de départ

La section précédente ayant permis de mieux appréhender l'influence de quelques paramètres sur les résultats obtenus, la suite la plus naturelle de notre étude nous a semblé être le relachement de quelques-unes de nos hypothèses de départ afin d'essayer de rendre notre modèle un peu plus réaliste.

Il nous est par exemple apparu que l'hypothèse du sol dur pouvait, suivant le phénomène étudié, se révéler un peu trop éloignée de la réalité. Nous avons donc réalisé de nouveaux calculs en considérant un sol meuble et donc, l'apparition possible d'un phénomène de « creusement ». Les résultats obtenus n'étant pas convaincants (instabilités et formes inattendues), nous ne les avons pas utilisés.

3.4 Résultats obtenus dans le cas linéaire

Nous avons ici utilisé le programme fourni par M. P.-Y Lagrée qui utilise la transformée de Fourier pour calculer le frottement.

Les résultats obtenus sont difficiles à comparer avec ceux obtenus avec *CASTEM*. En effet, lorsqu'on attribue à τ_S et Λ des valeurs du même ordre que celles utilisées avec *CASTEM*, les dunes évoluent rapidement vers des formes aberrantes (Fig 22). En outre, le choix de ces paramètres est difficile dans la mesure où la sensibilité de la solution à ces derniers est très grande.

Nous ignorons si ces mauvais résultats sont dus à des choix incorrects des différents paramètres du modèle ou s'ils sont la conséquence de la linéarisation des équations de Navier-Stokes.

On peut néanmoins souligner que lorsque l'on attribue à τ_S et Λ des valeurs très faibles, il est possible d'observer une dune qui avance mais sa déformation et son affaissement sont extrêmement lents (Fig 23).

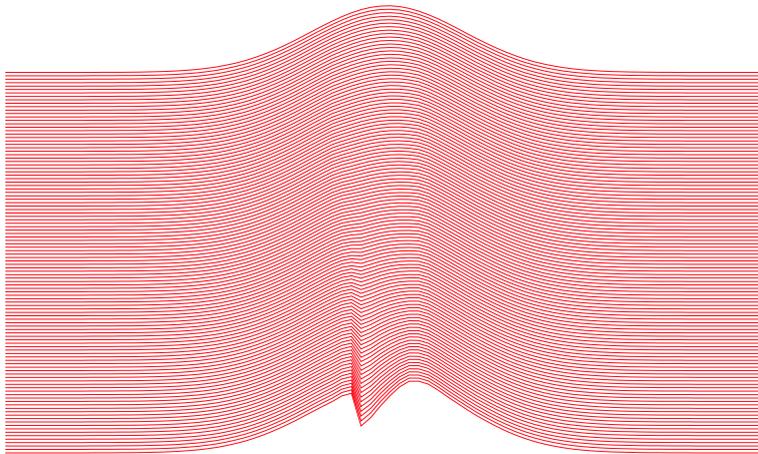


FIG. 22 – Evolution d'une dune dans le cas linéaire

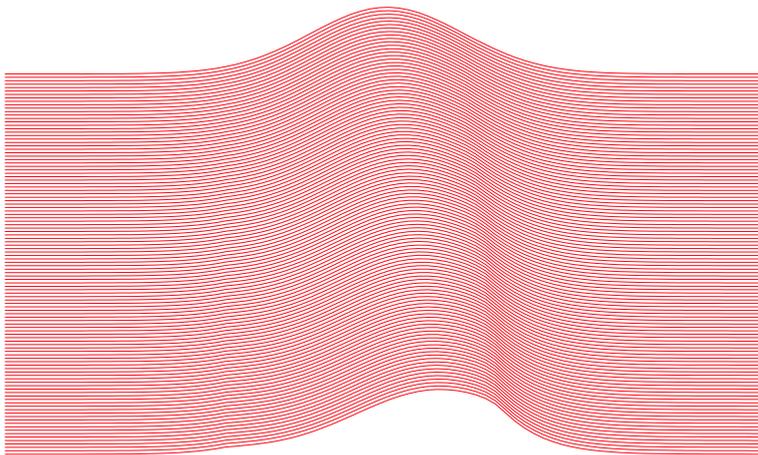


FIG. 23 – Evolution d'une dune dans le cas linéaire

Conclusion

Les résultats que nous avons obtenus au cours de cette étude sont intéressants. On peut cependant regretter la durée des temps de calcul nécessaire à l'observation d'une évolution significative d'une dune avec *CASTEM* (certains calculs ont nécessité plusieurs jours). On voit là tout l'intérêt d'utiliser le modèle linéarisé qui produit des résultats très rapidement bien que l'ajustement des paramètres soit plus délicat. Il faut toutefois garder à l'esprit que les résultats obtenus sont simplifiés et peuvent s'avérer insuffisamment précis.

Concernant les Travaux de Modélisation et de Simulation en général, la prise en main des logiciels s'est avérée délicate mais néanmoins formatrice. Nous regrettons de ne pas avoir pu disposer de plus de temps pour étudier plus en détails ce phénomène. Par exemple, il aurait pu être intéressant de se pencher sur le problème de l'évolution de plusieurs dunes car c'est un problème plus réaliste.

Annexe A : Programme C

```

#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include "math.h"

/*-----*/
float* x=NULL,* yw=NULL,* ywp=NULL;
float* p=NULL;
float* q=NULL;
float* qx=NULL;
float* Ar=NULL;
float* tau=NULL;

/*-----*/
double Pi;
double Lx,Lb,h0,V,taus,lambda;
double t,dt,dx,tmax;
double hmax,hmaxo,vhmax,xhmax,xhmaxo;
int it,nit;
int nx,type;

FILE *f,*g;
time_t sec;
time_t sec0;
int hrs,mins,secs;

/*-----*/
void trace(int ii);
void reLit(void);
void blabla();
void init();
void changesol(int ii);

/*-----*/
int main(void){
    int itt=0;

    // lecture des parametres
    reLit();
    Pi=4*atan(1);

    if(dt<0.0000001) nt=1;
    else nt=tmax/dt;

    t=0;

    f=fopen("profil.csv","w");
    fclose(f);

    f=fopen("xyt.OUT","w");
    fclose(f);

    hmax=h0*1.01;
    vhmax=0;
    // fin des lectures

    // RAZ
    init();

    // le calcul commence maintenant
    while(t<tmax){
        printf("Iteration %d, t=%f",itt,t);
        system("castem2001 dune.dgibi");
        changesol(itt);
        trace(itt);
        itt++;
        t=t+dt;
    }
    return 0;
}

/*-----*/
void trace(nit){
    static int i,icoul,e,ia,ixm;
    static float xp1,xp2,yp1,yp2,ym,xm,masse;
    static float hmaxmax=0;
    static char nom[18];
    int nmax;

    masse=0;
    for(i=1;i<=nx;i++) masse = masse + yw[i] * dx;

    hmax=0;
    nmax=0;
    ym=0;
    for(i=1;i<=nx;i++){
        if(yw[i]>ym){
            ym=yw[i];
            xm=x[i];
            ixm=i;
        }
    }
    printf("ym=%lf xm=%lf masse, vitesse)-%lf %lf \n",
        ym,xm,masse,0.25*(q[ixm]/yw[ixm]+q[ixm-2]/yw[ixm-2]
        +q[ixm-4]/yw[ixm-4]+q[ixm-6]/yw[ixm-6]));

    for (i=1; i<nx;i++){
        if(yw[i]>hmax){
            hmax=yw[i];
        }
        if(yw[i]>0){
            if((yw[i+1]-yw[i])*(yw[i]-yw[i-1])<0){
                nmax=nmax+1;
            }
        }
    }
    printf("t=%lf iteration=%d hmax=%lf
n (nombre d'extrema ?)=%d\n",t,nit,hmax,nmax);

    if (hmax>hmaxmax) hmaxmax=hmax;

    /* sauve */
    /*====sauvegarde profils instantannes====*/
    f=fopen("xyptq40.OUT","w");
    for (i=0; i<nx;i++){
        fprintf(f,"%f %f %f %f %f %f %f \n",
            x[i],yw[i],tau[i],-qx[i],Ar[i],ywp[i],
            (yw[i+1]-yw[i])/(x[i+1]-x[i]));
    }
    fclose(f);

    /*====sauvegarde profil de Dune====*/
    f=fopen("xyt.OUT","a");
    for (i=0; i<nx;i++){
        fprintf(fa,"%3.2f %4.5f %3.2f \n",x[i],yw[i],t);
    }
    fprintf(f,"\n");fprintf(fa,"\n");
    fclose(f);

    // fichier pour le film sous GNUPLOT
    f=fopen("mov.gnu","w");
    fprintf(f,"print k \n");
    fprintf(f,"p [-%lf:%lf] [0:%f]
'xyt.OUT' i k u 1:2t 'bosse' w l \n",Lx,Lx,hmaxmax);
    fprintf(f,"k=k+1 \n");
    fprintf(f,"if(k<%d) reread \n",nit); //,it/5);
    fclose(f);

    printf("it=%6d ===== film t=%lf\n",nit,t);

    /*====sauvegarde vitesse et masse====*/
    f=fopen("vitesse.OUT","a");
    fprintf(f,"%f %f %f \n",t,masse,
        (q[ixm-1]/yw[ixm-1]+q[ixm]/yw[ixm]+q[ixm+1]/yw[ixm+1])/3);
    fclose(f);

    /*====Affichage du tps====*/
    time(&sec);
    hrs=(sec-sec0)/3600;
    mins= ((sec-sec0)/60)%60;
    secs=(sec-sec0)%60;
    printf(" clock = %d h = %d min = %d s \n",hrs,mins,secs);
}

/*-----*/

```

```

void reLit(){
    g = fopen("parametres.in", "r");

    fscanf(g,"dt=%lf          \n",&dt);
    fscanf(g,"tmax=%lf       \n",&tmax);
    fscanf(g,"nx=%i         \n",&nx);
    fscanf(g,"Lx=%lf        \n",&Lx);
    fscanf(g,"Lb=%lf        \n",&Lb);
    fscanf(g,"h0=%lf        \n",&h0);
    fscanf(g,"type=%i       \n",&type);
    fscanf(g,"taus=%lf      \n",&taus);
    fscanf(g,"V=%lf         \n",&V);
    fscanf(g,"lambda=%lf    \n",&lambda);
    fclose(g);

    dx=2*Lx/(nx-1);

    if(!(dt<=0.1*dx)) dt=0.1*dx;
}

/*-----*/
void init(){
    double fx,fy,fa,qq,ym,yM,masse;

    float seed;
    int i;

    x = (float*)calloc(nx,sizeof(float));
    yw = (float*)calloc(nx,sizeof(float));
    ywp = (float*)calloc(nx,sizeof(float));
    tau = (float*)calloc(nx,sizeof(float));
    p = (float*)calloc(nx,sizeof(float));
    q = (float*)calloc(nx,sizeof(float));
    qx = (float*)calloc(nx,sizeof(float));
    Ar = (float*)calloc(nx,sizeof(float));

    /* Initialisation conditions initiales */
    xhmax=-0.5;
    for(i=0; i<nx;i++){
        x[i]=-Lx+ i*dx;
    }

    /* expon */
    if((type)==1){
        for(i=0; i<nx;i++){
            yw[i]=h0*exp(-Pi*pow((x[i]-xhmax)/Lb,2));
            qx[i]=0;
            Ar[i]=0;
        }
    }

    /* cos */
    if((type)==2){
        for(i=0; i<nx;i++){
            yw[i]=0;
            if(fabs(x[i]-xhmax)<(Lb/2)){
                yw[i]= h0*(1+cos( Pi*(x[i]-xhmax)/(Lb/2) ))/2;
            }
            qx[i]=0;
            Ar[i]=0;
        }
    }

    masse=0;
    for(i=1;i<nx;i++){
        masse = masse + yw[i] * dx;
    }

    // Ecriture de profil.csv
    f=fopen("profil.csv","w");

    fprintf(f,"%f %f\n", (float)(-Lx),0.);
    for(i=1;i<nx-1;i++){
        fprintf(f,"%f %f\n",x[i],yw[i]);
    }
    fprintf(f,"%f %f\n", (float) Lx,0.);
    fclose(f);

    /* fin init geometrie */

    printf(" Coefficients lus dans parametres.in\n");
    printf(" nbre de points = %d \n",nx);
    printf(" Lx = %7f (1/2 taille domaine) \n",Lx);
    printf(" Lb = %7f (1/2 longueur bosse) \n",Lb);
    printf(" dx = %7f (calculé 2*Lx/(nx-1)) \n",dx);

    printf(" dt = %7f (<.1 dx ?) \n",dt);
    printf(" tmax = %7f (tps final) \n",tmax);
    printf(" arrachement=(if)*(tau-(%5f)-%5f f')\n",taus,lambda);
    printf(" cas cisaillement pur \n");
    printf(" type de bosse initiale -> %d ",type);
    if((type)==1){printf("gaussienne de largeur Lb\n");}
    if((type)==2){printf("sinus\n");}
    printf(" masse initiale = %lf\n",masse);
    printf("[Return]\n");
    time(&sec0);
    printf(" clock = %d \n",sec0);
}

/*-----*/
void changesol(iterf){
    static int i,j,ixm,ityc;
    static float *yw,*Arm,*taucastem,*xcastem;
    static double taud,dqdx,Armm;
    int etat;
    char string [255];
    char * zaza;

    Arm = (float*)calloc(nx,sizeof(float));
    xcastem = (float*)calloc(10*nx,sizeof(float));
    taucastem = (float*)calloc(10*nx,sizeof(float));

    printf("Debut du calcul du nouveau sol...\n");

    g=fopen("cf.out","r");

    for(i=0;i<10*nx;){
        zaza = fgets(string,254,g);
        etat=(zaza==NULL);
        if(etat) break;
        etat=sscanf(string," %E %E ",&xcastem[i],&taucastem[i]);
        if(etat==EOF) break;
        if(etat==2){
            printf("i=%d, x=%f, tau=%f \n",i,xcastem[i],taucastem[i]);
            i++;
        }
    }

    fclose(g);

    j=0;
    for(i=0;i<nx;i++){
        while(xcastem[j]<x[i] && j<10*nx){
            j++;
        }
        if(xcastem[j]==x[i]) tau[i]=taucastem[j];
        else tau[i]=taucastem[j-1]+
            ((taucastem[j]-taucastem[j-1])/dx)*(x[i]-xcastem[j]);

    }

    /* Les deux lignes suivantes permettent de corriger
    les 2 valeurs extremales de taucastem qui sont mal calculées*/
    tau[0]=tau[1];
    tau[nx-1]=tau[nx-2];

    q[0]=0;

    /*---- Calcul du nouveau fond au pas de temps suivant ----*/
    /* seuillage du frottement, calcul de l'arrachement */
    Ar[0]=0;
    for(i=1;i<nx;i++){
        taud=tau[i] - lambda*(yw[i]-yw[i-1])/dx ;

        if(taud>taus) Ar[i]=V *(taud-taus);
        else Ar[i]=0;
    }

    /* valeur maximale admissible de l'arrachement */
    Armm=0.;
    for(i=1;i<nx;i++){
        Armm[i] = (1.+0.5*V*dx)*yw[i]/dt + V*q[-1+i] ;
        if(Ar[i]>=Armm[i]) Ar[i]=Armm[i];
    }

    /* calcul du flux de materiau q */
    /*
    d q
    --- + V q = Ar
    d x
    et

```

```

    d y      d q
    ---v = - ---
    d t      d x

*/
for(i=1;i<nx;i++){
    q[i]=(q[i-1] *(1/dx- 0.5 * V) + Ax[i])/(1./dx + 0.5 * V);
    qx[i] = (q[i]-q[i-1])/dx;
    yw[i] = yw[i] + (dt)*(-qx[i]);
}

// Mise a jour de profil.csv
f=fopen("profil.csv","w");
fprintf(f,"%f %f\n",(float)(-Lx),0.);
for(i=1;i<nx-1;i++){
    fprintf(f,"%f %f\n",x[i],yw[i]);
}
fprintf(f,"%f %f\n",(float) Lx,0.);
fclose(f);

free(Arm);
free(taucastem);
free(xcastem);

printf("Fin du calcul du nouveau sol\n");
}

```



```

* Associe au maillage un modele de comportement
* (Navier Stokes) et un type d'elements finis (macro)
$mt = mode mt 'NAVIER_STOKES' MACRO ;

* Imprime a l'ecran les informations sur le maillage
doma $mt 'IMPR' ;

$drc = mode dro 'NAVIER_STOKES' MACRO ;

mt2 = mt ;

* Recupere le maillage de base utilise pour la discretisation
mt = doma $mt maillage ;

* Nombre de Reynolds
Nu = 1./1000. ;
tos = 0. 0. ;

RV = EQEX $MT 'ITMA' nbit 'ALFA' 0.9
'OPTI' 'SUPG'
'ZONE' $MT 'OPER' CALCUL
'ZONE' $MT 'OPER' 'NS' NU 'INCO' 'UN'
'OPTI' 'CENTREE' 'EFM1'
'ZONE' $MT 'OPER' 'DFDT' 1. 'UN' 'DELTAT' INCO 'UN'
'ZONE' $DRO 'OPER' 'TOIM' tos 'INCO' 'UN' ;

* Impose les conditions aux limites

* Cisaillement
Umax=1.0;
Ucis=Umax*((coord 2 mt2)-(coord 2 p1))/(coord 2 p7);

RV = EQEX RV
'CLIM' 'UN' 'UIMP' gau Ucis
'UN' 'VIMP' gau 0.0
'UN' 'UIMP' bas 0.0
'UN' 'VIMP' bas 0.0 ;

RVP = EQPR $MT KTYPI 1
ZONE $MT 'OPER' 'PRESSION' 0. ;

RV.INCO=TABLE INCO ;

* 'UN' est un vecteur initialise a (Ucis 0.)
* et ayant pour origine les sommets du maillage
Ucis=NOMC 'UX' Ucis;
RV.INCO.'UN'=kcht $MT VECT SOMMET Ucis;
*unch = vect RV.INCO.'UN' 0.1 UX UY JAUNE ;
*TRACE UNCH MT2 'TITR' 'Champ de vitesse initial';

* idem
rv.inco.'UNM1' = kcht $MT 'VECT' 'SOMMET' (1.E-5 1.E-5 ) ;

rv.inco.'IT' = PROG 1 ;

rv.inco.'ER' = PROG 0. ;
rv.inco.'NBIT' = 0 ;
RV.PRESSION =RVP ;

REPETER bloc2;
EXEC RV;
err2 = (LOG err1)/(LOG 10.);
test=((EXTR RV.INCO.'ER' (&bloc2*nbit/5) < err2);
SI ( test);
QUITTER bloc2;
FINSI;
FIN bloc2;

*****
* Affichage des resultats
*****
unch = vect RV.INCO.'UN' 0.1 UX UY JAUNE ;
*TRACE UNCH MT2 'TITR' 'Champ de vitesse';

* Transforme les donnees sur la pression
pn = elno $MT RVP.PRESSION ;

* Trace la pression
*TRACE Pn mt (cont mt) 'TITR' 'Champ de pression';

evops=evol CHPO (coord 1 bas) bas;
evopn=evol CHPO Pn SCAL bas;

ev=evol JAUNE MANU (extr evops ordo)(extr evopn ordo);

*dess ev 'TITR' 'Pression sur le sol';

*****
* Calcul du coefficient de frottement *
*****
* On recupere les normales au bord
nn=doma $mt NORMALEV;
nn=NOMC (MOTS UX UY) (MOTS NX NY) nn;

* On recupere les composantes de la vitesse
vx=kcht $mt 'SCAL' 'SOMMET' 'COMP' 'UX' 0. rv.'INCO'.'UN';
vy=kcht $mt 'SCAL' 'SOMMET' 'COMP' 'UY' 0. rv.'INCO'.'UN';

* On calcule leur gradients (matrice)
grvx = kops vx 'GRAD' $mt; grvx = elno $mt grvx;
grvx=redu grvx (cont mt2);
grvy = kops vy 'GRAD' $mt; grvy = elno $mt grvy;
grvy = redu grvy (cont mt2);

* Calcul des composantes du vecteur gradient
* puis du vecteur gradient
gx=kops grvx 'PSCA' nn;
gy=kops grvy 'PSCA' nn;
gx=-1.0*gx;
gy=-1.0*gy;
gr= (NOMC 'UX' gx) ET (NOMC 'UY' gy);

* On forme les vecteurs tangents
tt = exco (MOTS 'NY' 'NX') nn (MOTS 'TX' 'TY');
mm = manu 'CHPO' (cont mt2) 2 'MX' -1.0 'MY' 1.0;
tt = kops tt '*' mm;
tt = NOMC (MOTS UX UY) (MOTS TX TY) tt;

* On determine enfin le coefficient de frottement
cf = kops gr 'PSCA' tt;
evocf=evol CHPO cf SCAL bas;
evcf=evol ROUGE MANU (extr evops ordo)(extr evocf ordo);
*dess evcf 'TITR' 'Coefficient de frottement sur la paroi !!!';

unch = vect RV.INCO.'UN' 0.2 UX UY JAUNE ;
*TRACE UNCH MT2 'TITR' 'Champ de vitesse !!!';

*****
*Sauvegarde du frottement (effacer les $ a la main)*
*****

x1=EXTR ( evcf ) ABSC;
y1=EXTR ( evcf ) ORDO;

np= DIME x1;
I=0;
OPTI IMPR 10 IMPR 'cf.out';
REPETER BCL1 (np);
I=I+1;
xi= EXTR x1 I;
yi= EXTR y1 I;
MESSAGE xi ' ' yi;
FIN BCL1;
OPTI IMPR 6;

fin;

```

Références

- [1] C.T. Yang (1995), *Sediment transport : theory & practice*, Mc Graw Hill
- [2] B. Andreotti, P. Claudin P., S. Douady (2002), *Selection of dune shapes and velocities. Part 2 : A two-dimensional modelling*, Eur. Phys. J., B28 , pp 341-352
- [3] K.K.J. Kouakou, H. Caps, P.-Y. Lagrée (2003), *Comparaisons numériques, analytiques et asymptotiques pour l'écoulement sur une dune sous marine*, Congrès Français de Mécanique, Nice 01-05 sept. 2003
- [4] K.K.J Kouakou (2003), *Erosion et sédimentation dans un canal*, Rapport partiel
- [5] Huerre P. (1998), *Mécanique des fluides, Tome 1*, chap. VIII et IX, Éditions de l'École Polytechnique
- [6] De Langre E. (2002), *Fluides et Solides*, Éditions de l'École Polytechnique