

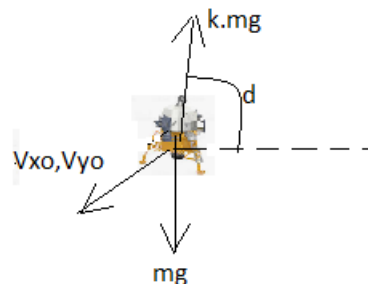
## TP algorithme génétique

Les algorithmes génétiques sont classiquement utilisés pour résoudre des problèmes d'optimisation. Fondamentalement, la recherche est locale. La solution est en effet trouvée par des déplacements élémentaires dans l'espace d'état. Le point fort de la méthode est d'effectuer de multiples recherches simultanées.

Nous allons appliquer cette méthode au problème suivant :

Un atterrisseur lunaire de masse  $m$  possède une position  $(x_0, y_0)$  et une vitesse  $(v_{x0}, v_{y0})$  par rapport au sol. Il dispose d'un moteur qui peut être orienté dans toutes les directions possibles et fournir une poussée variable. Toutefois, suite à une panne, il n'est plus possible de moduler la direction et l'intensité de la poussée, il faut les fixer une bonne fois pour toute et on ne pourra plus les modifier. L'objectif est néanmoins de le faire atterrir ( $y \leq 0$ ) avec une vitesse terminale  $(v_{xf}, v_{yf})$  aussi faible que possible, sachant qu'au-delà d'une certaine valeur de vitesse, on considère que le vaisseau va s'écraser. La question est donc : quelles valeurs choisir pour la direction initiale de la poussée et son intensité pour obtenir un freinage progressif et un atterrissage en douceur ? Une résolution analytique du problème est possible mais n'est pas demandée. L'objectif est de procéder à des simulations et de déterminer une solution acceptable à l'aide d'un algorithme génétique.

### Modélisation physique (pour information) :



Surface lunaire



On définit une poussée par :

- Une direction  $d$ , toujours vers le haut, donc entre  $0^\circ$  et  $180^\circ$ .
- Une force ou intensité  $f$ , qui est nécessairement supérieure au poids de l'engin, sinon, bien entendu, on va accélérer de plus en plus jusqu'à l'écrasement au sol. On définit cette intensité par  $k$  fois  $mg$ ,  $m$  étant la masse du vaisseau,  $g$  la constante de gravitation lunaire et  $k$  un réel supérieur à 1 et, selon les spécialistes, inférieur à 1,5.

Pour simplifier, on se place dans un plan 2D. On considère que la composante  $y$  pour la verticale est orientée vers le haut. A partir d'une position donnée  $(x_0, y_0)$  avec une vitesse donnée  $(v_{x0}, v_{y0})$ , les équations qui déterminent le mouvement sont les suivantes (on néglige la masse du carburant consommé) :

Accélération :

$$a_x = f \cos(d) / m = k g \cos(d)$$
$$a_y = (f \sin(d) - mg) / m = g (k \sin(d) - 1)$$

Vitesse en fonction du temps :

$$v_x(t) = a_x t + v_{x0}$$
$$v_y(t) = a_y t + v_{y0}$$

Position en fonction du temps :

$$x(t) = \frac{1}{2} a_x t^2 + v_{x0} t + x_0$$
$$y(t) = \frac{1}{2} a_y t^2 + v_{y0} t + y_0$$

(1)

**Application :**

On pourra prendre  $x_0 = 400$ ,  $y_0 = 400$ ,  $v_{x0} = -10$ ,  $v_{y0} = -10$ ,  $g = 1,6$

Les seuls paramètres du problème sont donc  $k$  et  $d$ . Un programme est fourni par l'enseignant. Il permet de demander à l'utilisateur de rentrer les valeurs de  $k$  et  $d$  manuellement et on affiche à l'écran si le vaisseau a atterri, ou si le résultat est un échec.

Explications (pour information seulement, il n'est pas nécessaire de comprendre ce passage pour faire le TP) : Pour certaines valeurs de  $k$  et de  $d$ , il n'est pas certain que l'engin atteigne  $y=0$ . Il faut vérifier que  $y(t)=0$  (équation (1)) admet une solution. Il s'agit d'une équation du 2<sup>nd</sup> degré. Si le déterminant est négatif, pas de passage par  $y=0$ , le vaisseau repart dans l'espace. S'il est positif, il y a 2 valeurs de  $t$  qui annule  $y$ . Si l'accélération verticale est négative (la poussée ne compense pas le poids), c'est la valeur de  $t$  la plus grande qu'il faut choisir (l'autre valeur de  $t$  est négative) et si l'accélération est positive, il y a bien un freinage et il faut alors choisir la valeur de  $t$  la plus petite (la 2<sup>ème</sup> correspond à un retour du vaisseau sur  $y=0$  après être passé sous la surface, ce qui n'a aucun sens). Appelons  $t_f$  la solution de cette équation et donc le moment où le vaisseau touche le sol. A partir de la valeur de  $t_f$ , on peut calculer les valeurs de vitesse  $v_x(t_f)$  et  $v_y(t_f)$ , comme indiqué dans les équations de vitesse. On admettra que l'atterrissage est réussi si et seulement si la norme de la vitesse est inférieure à 1 m/s.

Un programme C# vous est fourni. L'utilisateur peut modifier les valeurs de  $d$  et de  $k$ , puis en cliquant sur le bouton, le programme détermine et affiche si l'altitude 0 est atteinte et dans ce cas les valeurs des vitesses terminales  $v_x(t_f)$  et  $v_y(t_f)$  du module lunaire. Essayez d'atterrir manuellement, vous verrez, ce n'est pas facile ...

**Mise en œuvre de l'algorithme génétique :**

Principes généraux des algorithmes génétiques : comme pour l'évolution des espèces selon la théorie de Darwin, l'objectif est de créer une population nombreuse de vecteurs (code génétique), puis de sélectionner les meilleurs selon un critère d'évaluation donné, de régénérer une population nombreuse en procédant à des « mutations » des meilleurs vecteurs, puis d'effectuer à nouveau une sélection et ainsi de suite jusqu'à que les vecteurs finaux soient « adaptés ».

Pour déterminer le meilleur vecteur, nous avons besoin d'une méthode d'évaluation. En voici une. Soit  $v_i$  un vecteur ( $d_i$ ,  $k_i$ ). Si l'altitude 0 n'est jamais atteinte, on attribue une note arbitraire de 100000. Si l'altitude 0 est atteinte, conduisant in fine à la position ( $x_{fin}$ , 0) avec la vitesse ( $v_{xfin}$ ,  $v_{yfin}$ ), la note attribuée est :

$$\text{Note } (v_i) = v_{xfin}^2 + v_{yfin}^2$$

Plus la note est faible, meilleur est le vecteur.

**Algorithme génétique :**

- 1) Création d'une « population » de  $n$  vecteurs évaluable. Un vecteur évaluable est défini par :
  - une direction de poussée  $d$
  - un coefficient de poussée  $k$ .
  - une note (voir plus loin)

Une population de vecteurs est typiquement une liste. On prendra  $n$  assez grand, par exemple  $n=1000$ . Pour créer ces vecteurs aléatoires, on peut procéder de la façon suivante :

On définit la direction de poussée  $d$  par un tirage aléatoire d'un entier entre 0 et 180 et le coefficient multiplicateur de poussée  $k$  par un réel entre 1 et 2.

- 2) Evaluation des  $n$  vecteurs de la population. Pour chacun des  $n$  vecteurs, on effectue la simulation de la descente, on détermine la position et la vitesse terminales et on procède à l'évaluation, ce qui permet de déterminer la note.
- 3) La population doit ensuite être triée en fonction de la note, du plus petit au plus grand.
- 4) Sélection. On élimine de la population les plus mauvais (note la plus forte) et on ne garde que les meilleurs, par exemple les 100 premiers.
- 5) Mutation. L'objectif est de compléter la liste pour revenir à  $n$  vecteurs en ajoutant des vecteurs qui sont des copies des meilleurs, à l'exception de quelques modifications mineures. Pour créer un nouveau vecteur, on prend au hasard un des vecteurs restants et on le duplique. Mutation : on choisit une des propriétés du vecteur, soit la direction, soit la poussée de manière aléatoire et on modifie la valeur

également de manière aléatoire entre -3% et +3%. On ajoute le mutant à la liste et on recommence le processus jusqu'à ce qu'on obtienne à nouveau  $n$  vecteurs.

6) Aller en 2).

On peut itérer ce processus un certain nombre de fois (10 fois par exemple), puis afficher le meilleur vecteur de la population en donnant la position et la vitesse terminales.

**NB : Une partie du code est déjà écrite, il ne vous reste plus qu'à compléter !**

#### **Amélioration de l'algorithme :**

Testez différentes méthodes :

- Plutôt que de modifier la direction ou la poussée de manière aléatoire, on peut privilégier les petites modifications, par exemple comprises entre 0 et 10%.
- On peut également effectuer des modifications aléatoires de plus en plus petites au fur et à mesure qu'on fait évoluer les populations.
- On peut agir sur la population de vecteurs qui reste vivante.
- On peut aussi réduire ou augmenter la population totale.

#### **Croisement génétique :**

Plutôt qu'une mutation, on peut opérer un croisement génétique. Pour cela, il suffit de choisir 2 vecteurs différents dans la liste des vecteurs restants et de composer un vecteur mutant en prenant la direction de l'un et le coefficient de poussée de l'autre, ou réciproquement. Toutes les transformations génétiques ne doivent pas être des croisements. Le mieux est de panacher avec un pourcentage minimum de mutations.