Desaedeleer Maxime
Ruiz Arenas Nadia
Groupe 60

2ième Candidature en Ingénieur Civil

Année académique 2003-2004

Projet Matlab:

Le javelot

Desaedeleer Maxime
Ruiz Arenas Nadia

2/4/2004

Mécanique Rationnelle : Projet Matlab :

Etude du mouvement du javelot

Motivations

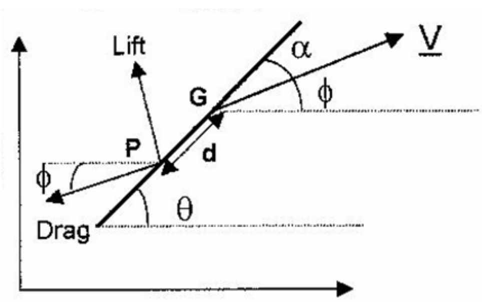
Outre la technique du lanceur, la position du javelot lors de son lancement et la manière dont il est jeté sont des éléments essentiels pour augmenter la distance entre le point de départ et le point d'impact avec le sol.

En effet, différents paramètres peuvent intervenir lors du lancement du javelot : l'angle d'attaque, l'angle de lancement, et la rotation que peut lui donner le lanceur.
Pour observer l'effet de ces paramètres et pour maximiser la portée du javelot, nous devons modéliser notre problème puis trouver et résoudre les équations du mouvement du javelot. Malheureusement, résoudre analytiquement de telles équations serait bien difficile, c'est pourquoi les méthodes d'analyse numérique et les programmes de simulation sont inhérents au métier d'ingénieur.

Ainsi, dans ce projet, nous mettrons en évidence les avantages que présente l'utilisation du logiciel Matlab dans la détermination des valeurs des différents paramètres qui maximise la portée du javelot.

Modélisation

Les différentes forces agissant sur le javelot peuvent être modélisées par la force de pesanteur et des forces proportionnelles au carré de la vitesse de son centre de gravité. Ces forces (excepté la force de pesanteur) sont représentées sur le schéma ci-dessous, ainsi que certains paramètres.



α = angle d'attaque [radian]
Normes des forces :
Lift = $K_L.V^2$ [N]
Drag = $K_D.V^2$ [N]
$K_L = 0{,}0127.\alpha^{1,34}$ [m^{-1}]
$K_D = 0{,}00024.e^{5,157.\alpha}$ [m^{-1}]

Ce solide possède trois degrés de liberté. Nous devons donc choisir 3 coordonnées pour pouvoir déterminer sa position dans l'espace. Prenons la position (x,y) du centre de gravité G, et l'angle θ.

Les équations du mouvement

Deux méthodes nous permettent de trouver les équations du mouvement :

-les équations de Lagrange :

$$\frac{d}{dt}\frac{\partial T}{\partial \dot{q}_i} - \frac{\partial T}{\partial q_i} = Q_i$$

-les théorèmes généraux :

$$\frac{d}{dt}\overline{R} = \sum_{i=1}^{n}\overline{F_i}$$

$$\frac{d}{dt}\overline{M_A} = \sum_{i=1}^{n}\overline{m_{A,i}}$$

Nous avons vérifié que les équations de Lagrange et les théorèmes généraux mènent aux mêmes équations, ce qui est rassurant. Nous n'allons donc développer que les calculs avec les théorèmes généraux.
Nous nous placerons dans un repère fixe Oxy (x étant l'axe horizontal et y l'axe vertical). Les coordonnées du point G seront (x,y).

Ainsi, calculons les différents éléments :

-la résultante cinétique : $\overline{R} = m.\overline{v_G} = m.(\dot{x}\overline{1_x} + \dot{y}\overline{1_y})$

-le moment cinétique : $\overline{M} = \overline{\overline{I}}_G.\overline{\varpi} = I.\dot{\theta}\overline{1_z}$

-les forces extérieures : $\sum_i \overline{F_i} = (-L.\sin(\phi) - D.\cos(\phi))\overline{1_x} + (L.\cos(\phi) - D\sin(\phi) - g)\overline{1_y}$

-les moments extérieurs en G : $\sum_i \overline{m_{G,i}} = -d(L.\cos(\alpha) + D.\sin(\alpha))\overline{1_z}$

Injectons les dans les formules des théorèmes généraux et nous obtenons ce système :

$$\begin{cases} \ddot{x} = -\dfrac{1}{m}(D.\cos(\phi) + L.\sin(\phi)) \\ \ddot{y} = -g + \dfrac{1}{m}(L.\cos(\phi) - D.\sin(\phi)) \\ \ddot{\theta} = -\dfrac{d}{I}(L.\cos(\alpha) + D.\sin(\alpha)) \end{cases} \quad \text{avec} \quad \begin{cases} \phi = arctg(\dfrac{\dot{y}}{\dot{x}}) \\ \alpha = \theta - \phi \end{cases}$$

Si nous parvenons à résoudre ce système, alors nous aurions les équations du mouvement. Mais cette tâche s'avère si ardue que nous préférons laisser ce travail Matlab qui nous représentera la trajectoire du javelot, ainsi que bien d'autres informations.

Résolution « numérique »

Nous devons faire appel à des méthodes d'analyse numérique pour résoudre ce système d'équations différentielles. Matlab possède dans ses fonctions de base la méthode « ode45 » qui résout des systèmes d'équations différentielles du premier ordre. Cependant, notre système d'équations est du deuxième ordre, nous devons donc recourir à une nouvelle présentation de ces équations.

En effet, ce système doit être mis sous forme de fonction dans Matlab pour que la méthode « ode45 » puisse nous fournir son intégrale. Mais ce résultat ne nous donne donc que l'estimation de la vitesse des coordonnées, c'est-à-dire la dérivée des valeurs que nous cherchons, à différents instants.
Pour résoudre ce problème, nous avons simplement introduit, dans ce même système, trois équations supplémentaires, qui sont l'évaluation de la vitesse des variables x, y, et θ.

En conclusion, nous obtenons la valeur des trois coordonnées généralisées ainsi que leur dérivées tout au long du mouvement du javelot et ce pendant un laps de temps déterminé. Cependant, nous devons encore présenter ces résultats via de petits traitements (recherche de maximum, graphiques,…).

Nos programmes Matlab

Plusieurs programmes Matlab sont en annexe à ce rapport, et ce afin de mettre en évidence les résultats et l'influence des différents paramètres sur ce problème.

Le méthode de résolution des équations est rigoureusement la même pour chaque programme. En fait, seules la présentation des résultats et la variation des paramètres diffèrent, et par conséquent, les lignes de code sont plus nombreuses et compliquées. Voici un descriptif de chaque programme :

-« javelot.m » : il fournit la distance atteinte par le javelot et affiche la trajectoire du centre de gravité. La valeur des paramètres est fixée dans ses lignes de code, il faut donc modifier le programme et le sauver pour faire différents essais, mais c'est le plus simple.

-« javelotangles.m » : c'est une fonction qui reçoit en entrée toutes les valeurs de l'angle thêta (angle que fait le javelot avec l'axe horizontal lors de son lancement) et qui présente comme sorties la distance maximale atteinte par le javelot et l'angle θ qui y correspond. Elle affiche également un graphe de la portée du javelot en fonction de θ.

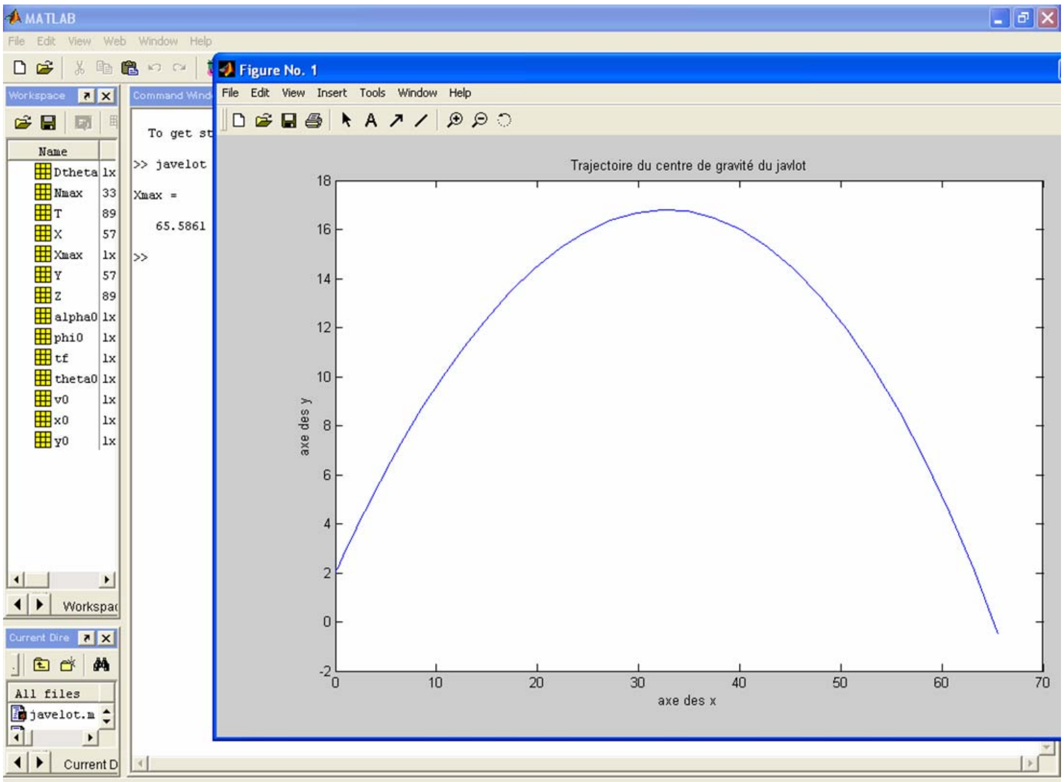
-« javelotanglesA.m » : cette fonction est identique à la précédente, mais l'angle θ reste fixe, et on traite le problème pour différents angles d'attaque (α) introduits comme entrée de la fonction.

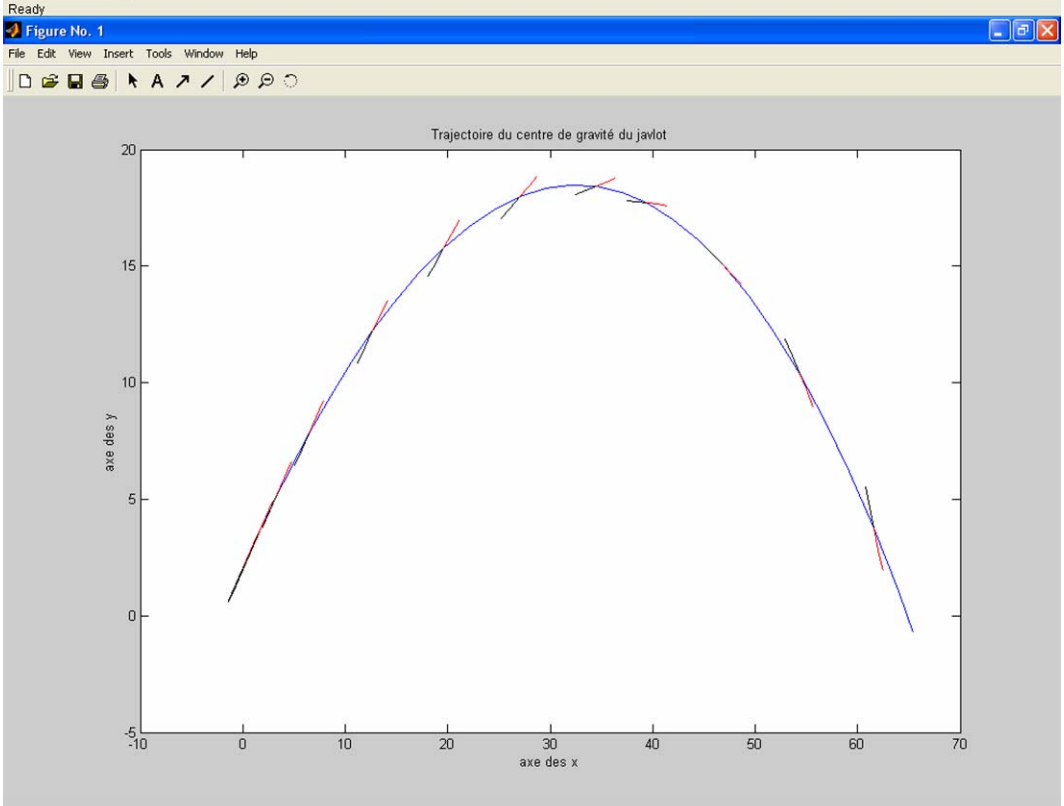
-« javelotangleAP.m » : elle reprend le code de « javelotanglesA.m », mais c'est l'angle de lancé φ qui reste constant.

-« javelotanglesTotal.m » : Analyse la porté maximale pour des angles θ et α variables. Cette fonction nous fournit la valeur des paramètres qui maximise la portée.

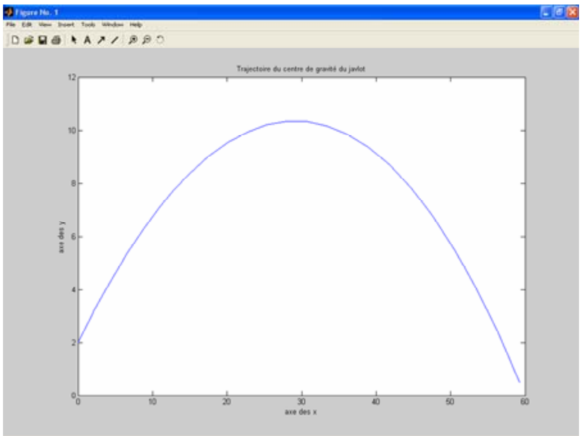
Présentation des résultats

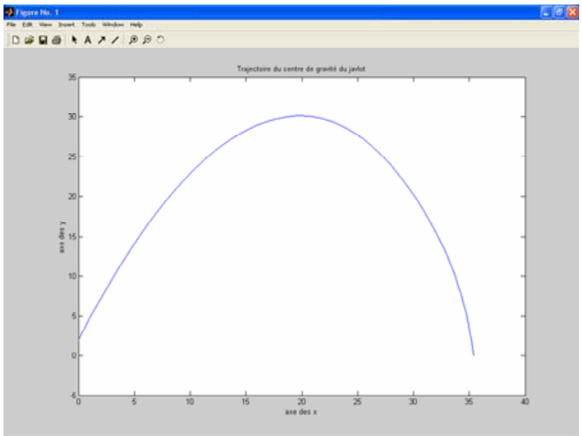
Pour pouvoir observer les effets des différents paramètres, il faut d'abord s'assurer que la résolution des équations du mouvement est cohérente. Pour ce faire, nous avons observé plusieurs cas avec le premier programme. Il nous fournit la trajectoire du javelot, et on peut aussi le faire apparaître sur le graphe à plusieurs endroits.





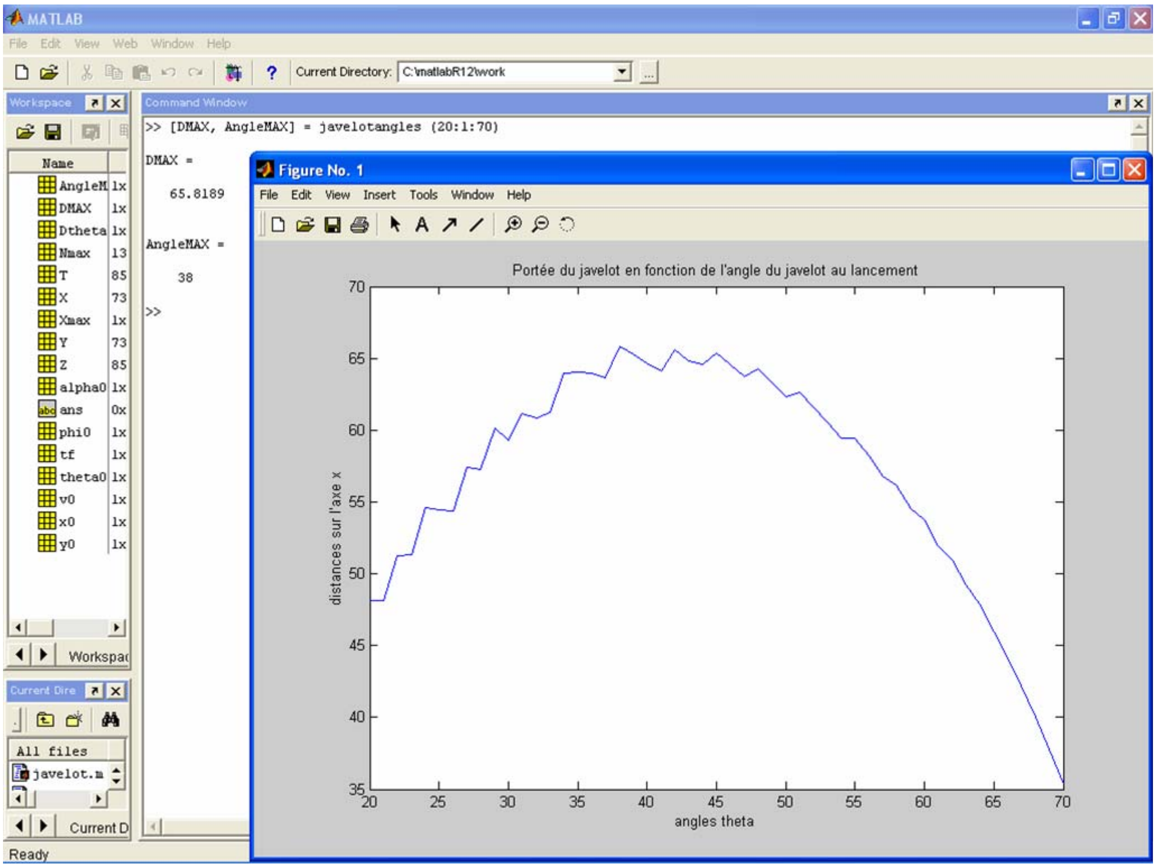
A première vue, la trajectoire ressemble à un tir parabolique, mais observons deux angles limites, par exemple 30° et 70° pour les valeurs de thêta (sa dérivée et alpha sont nuls).





Mais le lancé à 70° nous démontre le contraire par sa dissymétrie.

Avec un angle d'attaque choisi nul, on peut facilement trouver l'angle thêta optimal pour le lancé du javelot avec le deuxième programme :



On aurait pu s'attendre à ce que la courbe soit lisse, mais ces « irrégularités » dans la courbe résultent du fait que les forcent qui s'exercent sur le javelot varie suivant sa pénétration dans l'air. Nous avons observé précédemment que la vitesse du javelot n'était pas toujours parallèle à celui-ci durant sa trajectoire. Ainsi, pour deux angles pourtant très proches, cet angle d'attaque peut varier différemment et modifier la portance du javelot.

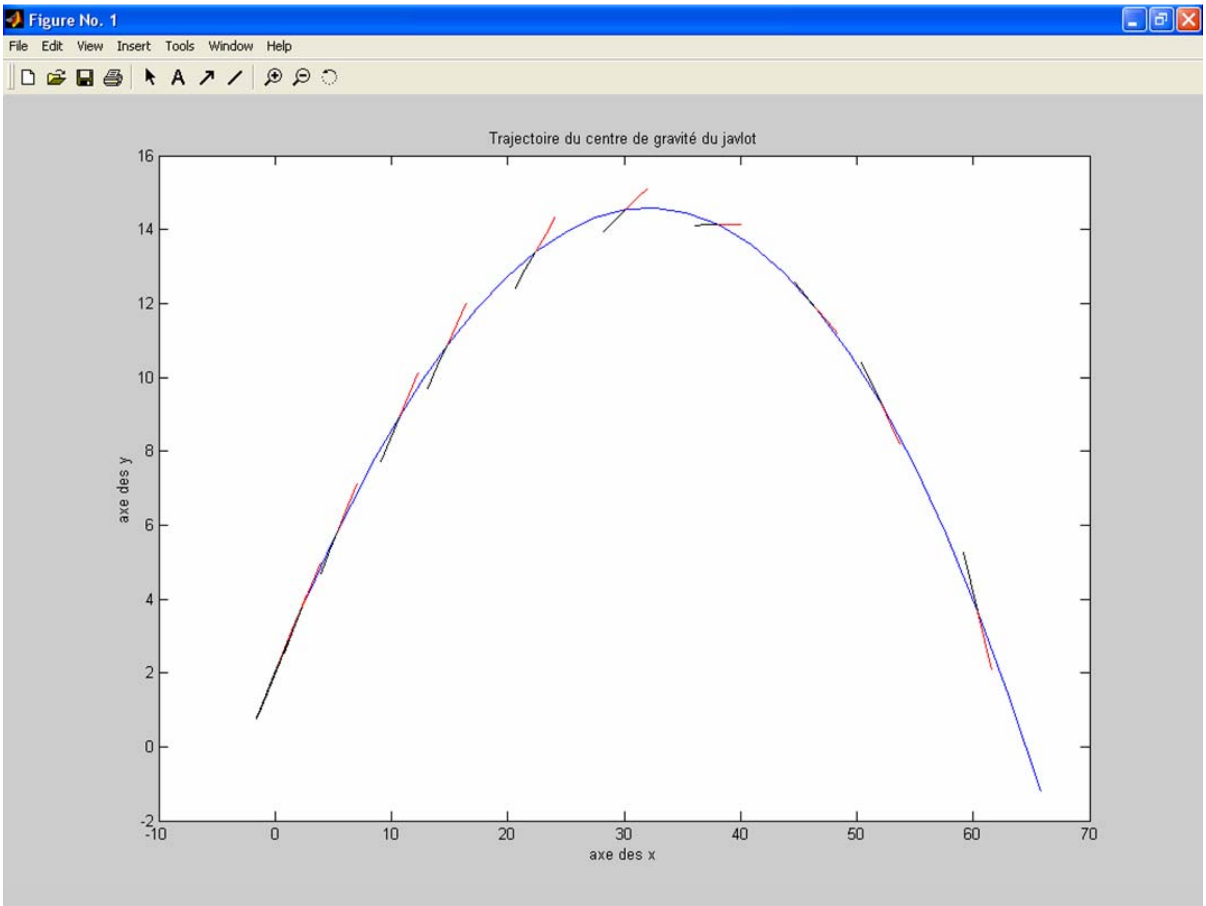


Figure No. 1
File Edit View Insert Tools Window Help
Trajectoire du centre de gravité du javlot
axe des y
axe des x
16
14
12
10
8
6
4
2
0
-2
-10
0
10
20
30
40
50
60
70

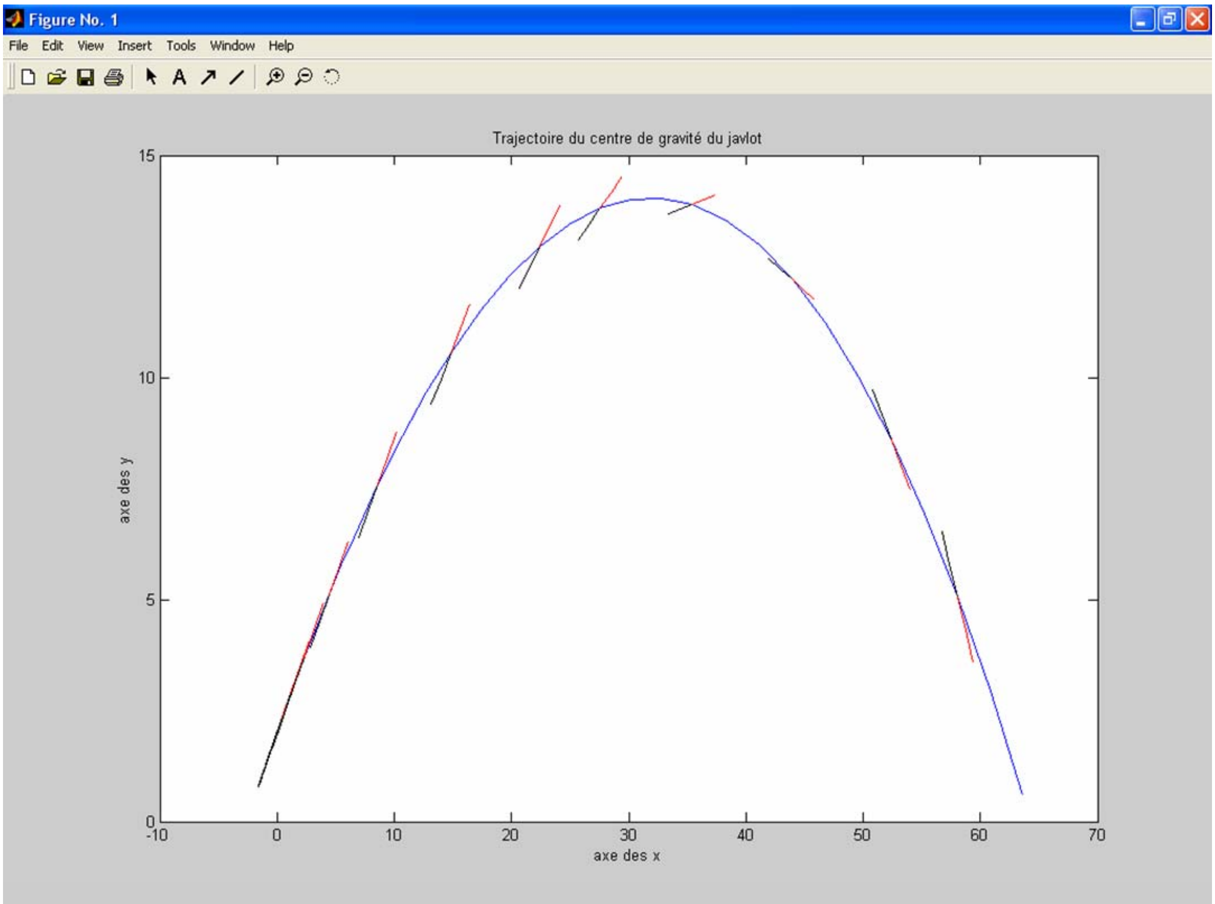
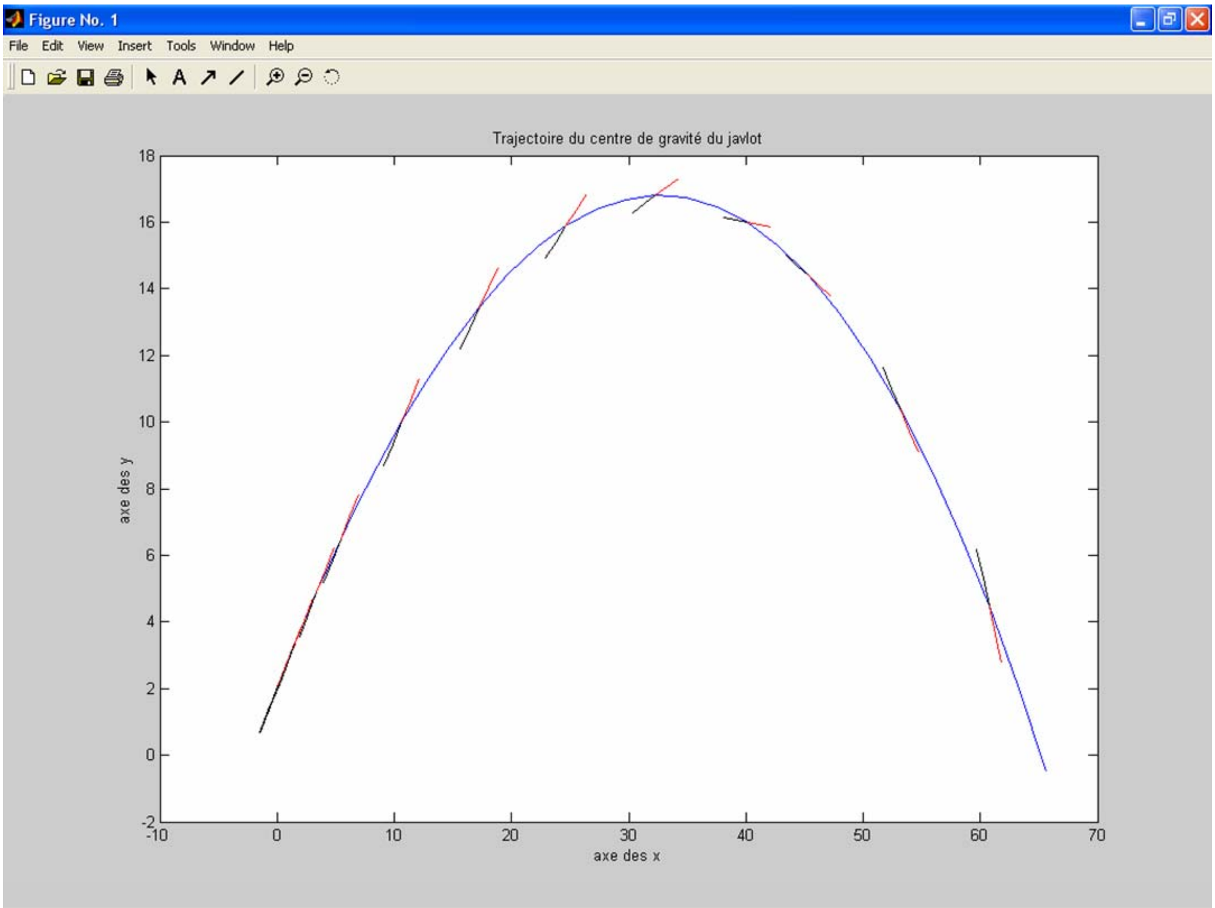


Figure No. 1
File Edit View Insert Tools Window Help
Trajectoire du centre de gravité du javlot
axe des y
axe des x
15
10
5
0
-10
0
10
20
30
40
50
60
70

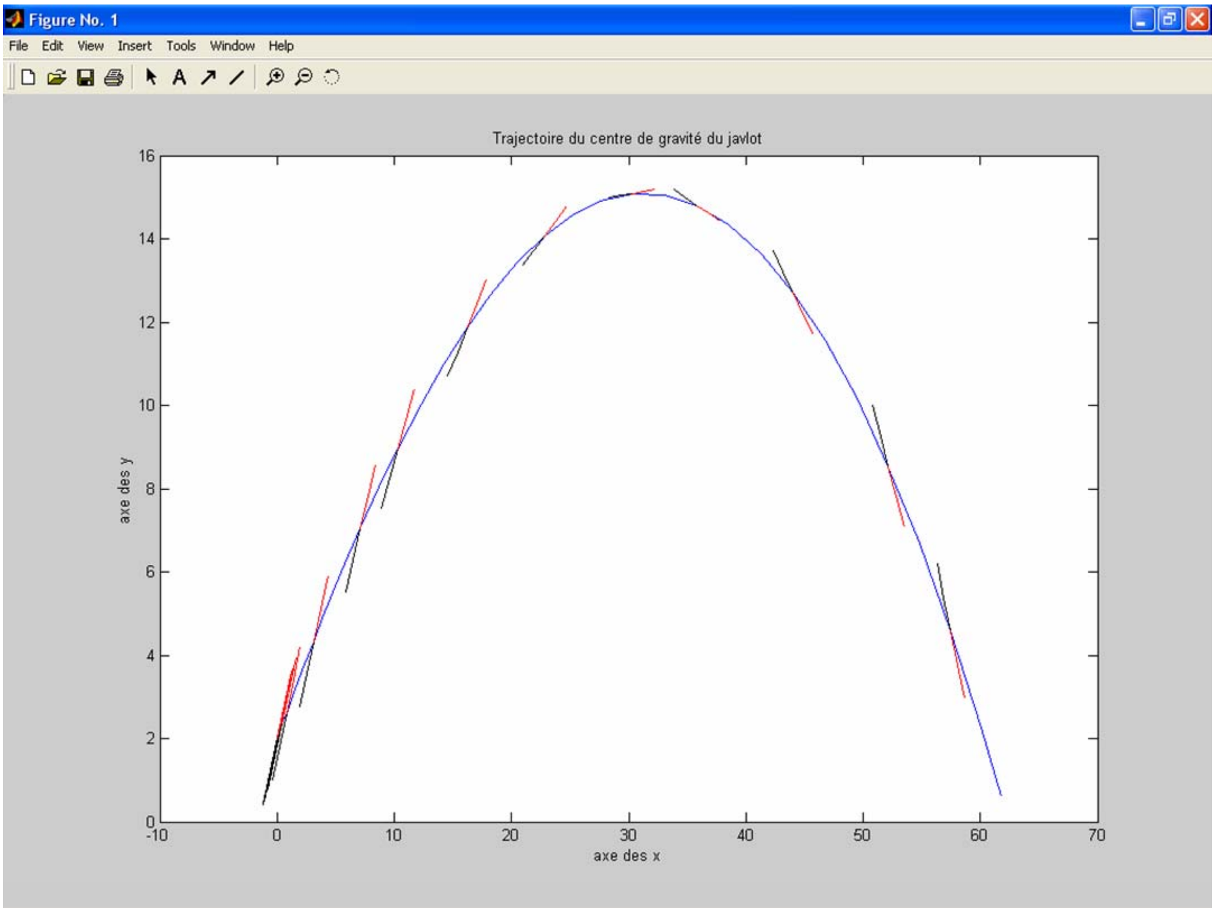
Le premier graphe illustre les positions du javelot lors d'un lancé à 38°, son angle optimal. Prenons le comme référence pour analyser le second graphe où le javelot est lancé avec une inclinaison de 37°. La portée passe de 66 mètres à près de 63 m. Or les angles sont très proches, et, jusqu'au sommet de la trajectoire, le javelot a le même comportement dans les deux cas. Mais nous remarquons que les premières et deuxièmes positions du javelot après le sommet diffèrent dans les deux graphes : alors que, dans le premier, le javelot a plutôt tendance à s'aligner avec sa vitesse et sa trajectoire, dans le second, il est plus lent à faire de même. Or les forces de frottements sont liées à cet angle, et cette petite perte de vitesse peut expliquer pourquoi les points d'impacts sont si éloignés.

Cette explication est d'autant plus cohérente que nous pouvons tenir le même raisonnement pour deux angles différents, mais qui sont des maxima locaux. Par exemple, si nous nous basons sur un angle thêta de 42°, nous remarquons que la portée varie peu par rapport à 38°, ce qui est dû à un comportement quasi identique du javelot lors de son mouvement, comme l'atteste le résultat fourni par Matlab :



Ces quelques exemples nous montrent que l'angle d'attaque alpha a une influence direct et très complexe sur le comportement du javelot. Un de nos buts est de déterminer l'angle d'attaque qu'il faut donner au javelot lors de son lancé pour atteindre une distance la plus grande possible. Nous savons que cet angle d'attaque est à l'origine d'une force qui « pousse le javelot vers le haut », c'est-à-dire qui modélise la portance du profil du javelot, mais cette inclinaison augmente les frottements car la pénétration dans l'air est moins bonne.
Le « juste dosage » entre ces deux forces déterminera quel angle sera nécessaire pour maximiser la porté du javelot.

Observons le comportement du javelot penché à 53° et lancé à 38° (alpha = 15°) :



Notre première constatation est que la portée a malheureusement diminué. Donc malgré la portance plus importante au lancé, le résultat n'est pas meilleur qu'avant. Ce résultat ne parait pas évident, mais si nous observons la position du javelot tout au long de sa trajectoire, nous ne remarquons aucun avantage à le lancer de cette manière dans la première moitié de la courbe : la portance ne compense pas les forces de frottement, et aucune amélioration n'est apportée. Par contre, à partir du sommet, le comportement est totalement différent. Le javelot bascule beaucoup plus tôt sur lui-même, et l'angle d'attaque devient très vite nul, puis négatif, avec pour conséquence une forte diminution de la portance et l'apparition plus rapide des forces de frottements.

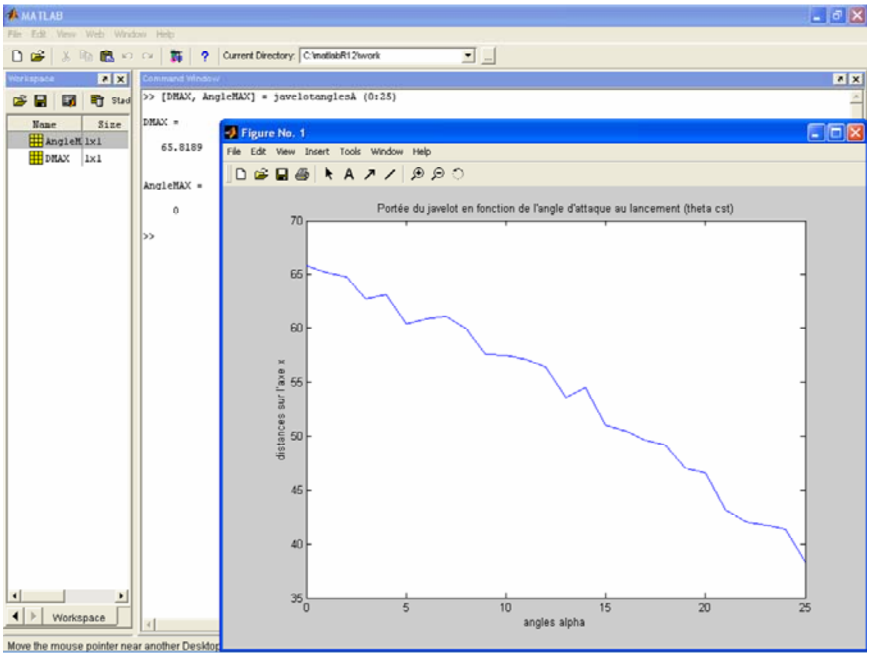
Ces différentes interprétations nous permettent de déjà dresser le comportement que devrait avoir le javelot.

Au début, nous devons lancer le javelot de manière à minimiser les frottements. Pour cela, nous devons réduire au maximum l'angle d'attaque (peut-être qu'un petit angle d'attaque puisse fournir un bilan portance - frottements positif, mais seule la simulation peut éclaircir ce point). Le vecteur vitesse va évoluer de manière à rester tangent à la trajectoire, alors que le javelot, d'après les essais réalisés jusqu'ici, a tendance à garder son inclinaison initiale jusqu'au sommet. Cet angle entre la vitesse et le javelot va avoir un effet très important sur la seconde partie de la trajectoire.

En effet, nous constatons que plus l'angle d'attaque est élevé lors de l'amorce de la descente, plus la portée est grande. Si nous revenons au bilan de force, nous pouvons justifier ce phénomène par le simple fait que la force « Lift », la portance, freine la chute du javelot, et comme la vitesse est minimale et horizontale, les forces de frottement n'ont que très peu de répercussions sur la trajectoire.

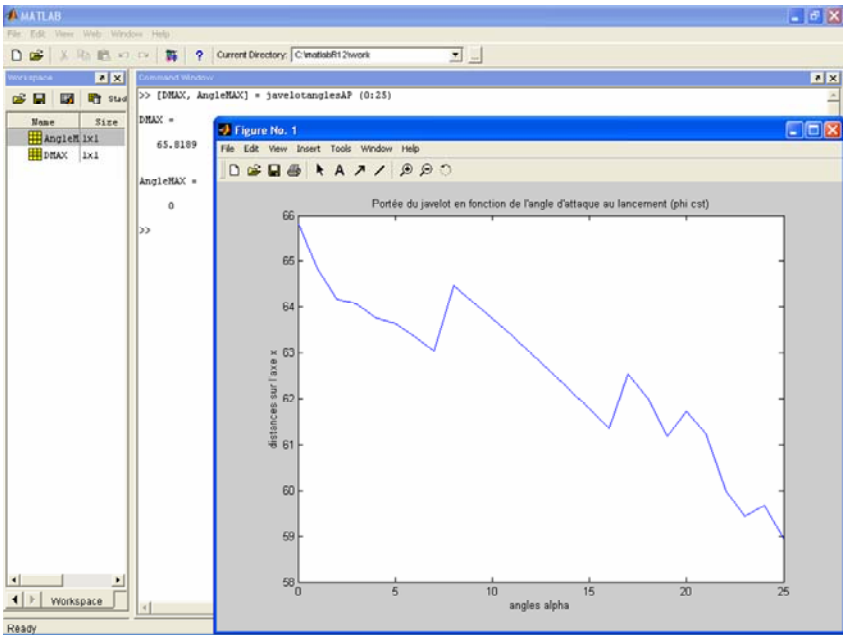
Pour la suite du mouvement, nous constatons qu'à chaque lancé, le javelot continue son mouvement de basculement dû à son inertie. Et que, dans les cas les plus favorables, le javelot conserve longtemps un angle d'attaque positif, mais il termine à chaque fois sa course avec un angle négatif. Et cet angle négatif est à éviter le plus possible car il n'engendre que des forces de frottements.

Pour comparer les résultats que produisent différents angles d'attaques lors du lancé du javelots, nous utiliserons nos deux programmes adéquats.
Le premier fait varier l'angle d'attaque (α) pour une inclinaison (θ) donnée et fixe du javelot lors de son lancé. Le second conserve l'angle de lancé/de la vitesse (φ) constant pour différents alpha (allant de 0° à 20°).



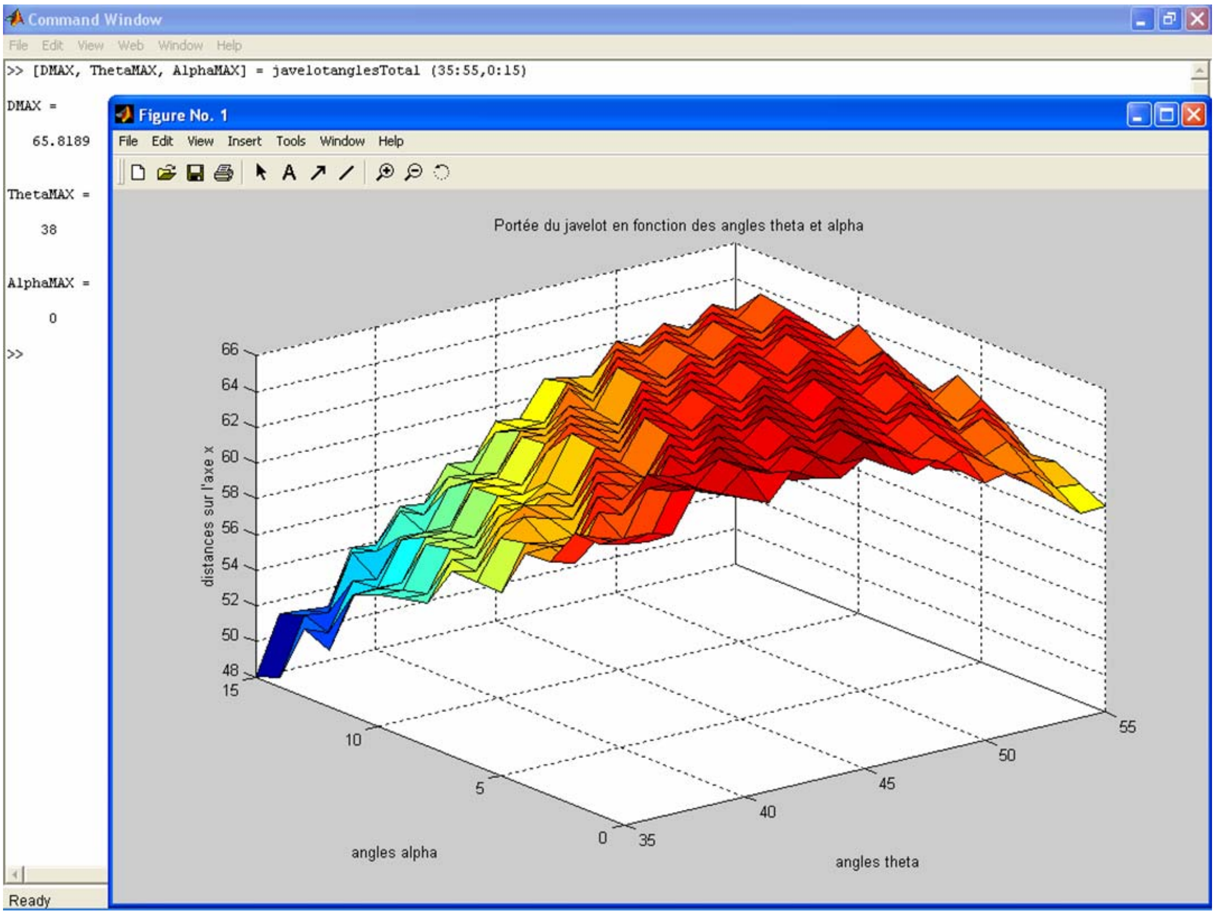
La distance maximale est atteinte avec un angle thêta de 38° pour la valeur de alpha égale à 0°.

Avec la simulation qui maintient l'angle de la vitesse constante, l'angle d'attaque initial qui maximise la portée est nul. Et nous retrouvons le cas de figure ci-dessus, et le même point d'impact.



Tous ces résultats sont en parfait accord avec les interprétations physiques que nous avions précédemment faites pour des angles d'attaque de 0° et de 15°.

Enfin, nous ne pouvons nous contentez de ces résultats, car nous avions, pour chaque expérience, fixé un paramètre (θ ou φ). Pour être complet, il convient de s'assurer qu'aucun autre comportement que ceux observés jusqu'à présent n'apparaissent pour d'autres valeurs de paramètres.
En utilisant la dernière fonction, nous pouvons couvrir un vaste domaine dans les valeurs de thêta et alpha. Le résultat est un graphique à trois dimensions, portant la portée du javelot en fonction de θ et α. Sa représentation est plus difficile, mais tous les éléments nécessaires à sa compréhension ont été traités avec les programmes précédents, et quelques lignes de codes permettent d'afficher dans Matlab la distance maximale ainsi que la valeurs des paramètres propres à ce lancé. Et voici les résultats :



En conclusion, nous retrouvons toujours les mêmes valeurs que précédemment. Mais nous observons clairement sur le graphe en 3D que la forme globale est prévisible : « c'est aux alentours de 40° - 50° et avec un angle alpha pas trop élevé que la portée est maximale, après cela diminue. » Mais cet aspect de surface, avec de nombreux maxima locaux (pics), nous confirme que ces résultats ne sont pas aussi intuitifs que nous aurions pu le croire, et justifie l'analyse simultanée des deux paramètres, car plusieurs couples de valeurs différentes fournissent de très bons résultats.

La fin d'une analyse ?

Le dernier point que nous devons encore traiter à ce sujet est la possibilité d'imprimer un mouvement de rotation au javelot lors de son lancer, c'est-à-dire ne pas considérer la vitesse de variation de l'angle thêta nul.

Nous pensions qu'un dθ/dt non nul ne changerait rien aux résultats précédents, et ce fut confirmé après avoir fait un essai avec les paramètres optimaux trouvés pour un dθ/dt nul (θ = 38°, α = 0°). En effet,si dθ/dt n'est pas nul, la distance est bien plus petite. Mais par curiosité, nous avons observé ce qui se passait si, pour plusieurs θ fixés, dθ/dt et α variaient simultanément. Pour 38°, dθ/dt et α sont nuls pour une distance maximale. Mais si nous utilisons une autre valeur de θ, assez différente de 38°, les valeurs des paramètres qui maximisent la portée ne sont plus nuls, et, plus intéressant, pour un θ choisi au hasard, la distance atteinte est plus grande que précédemment.

Nous avons donc développé quelques programmes supplémentaires :

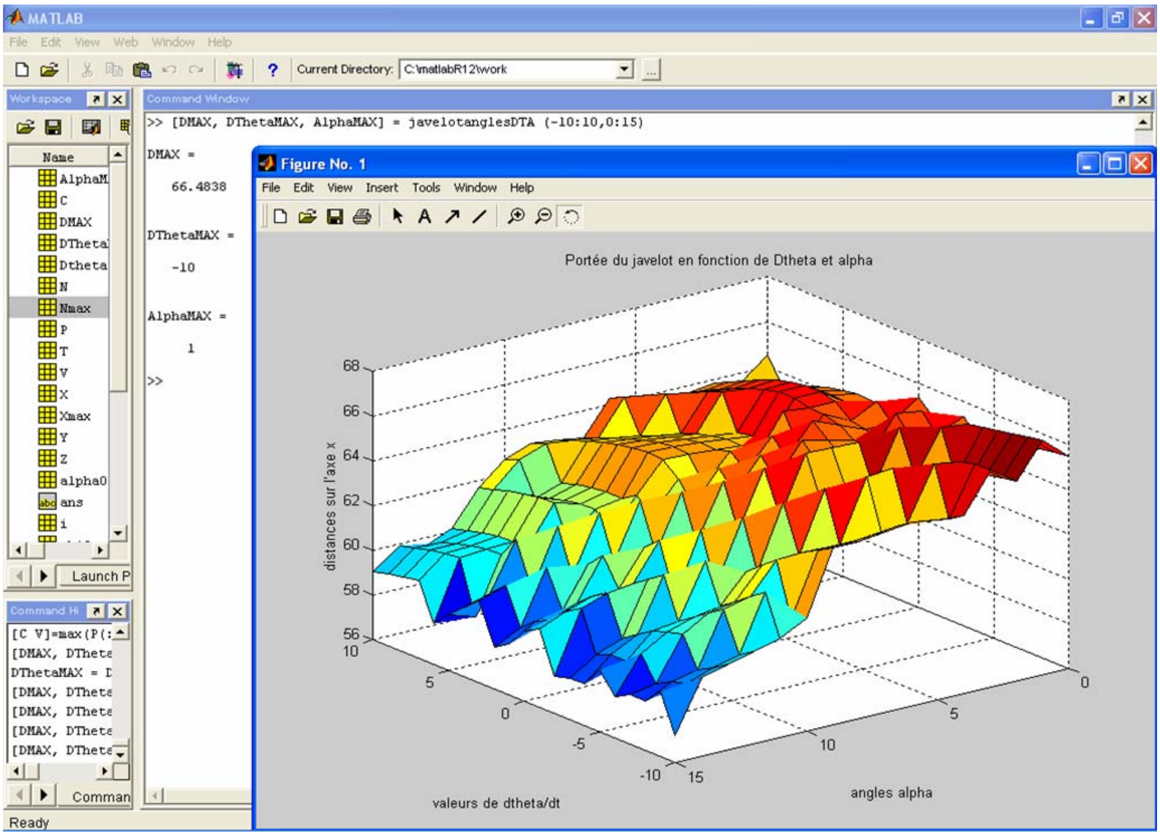
-« javelotanglesDT .m» : Cette fonction détermine le dθ/dt optimal pour θ et α fixés.

-« javelotanglesDTA.m » : Elle admet un seul angle θ, mais fait varier α et dθ/dt suivant des limites définies en entrées de la fonction.

-« javelotanglesTDTA.m » : Ce programme reçoit en entrées les valeurs limites de chaque paramètre, elle détermine ensuite les valeurs qui optimisent la lancé, et la distance atteinte.

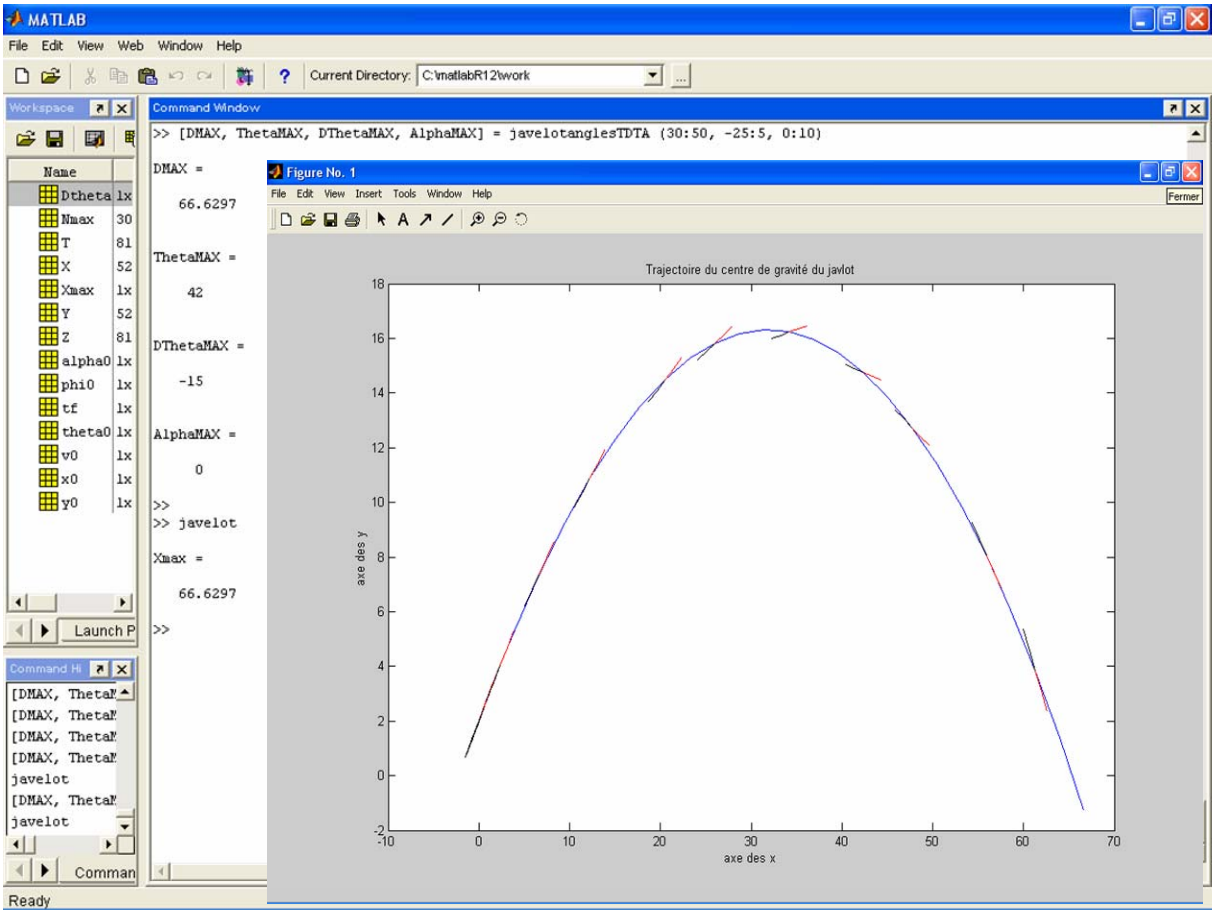
Nous allons donc présenter les résultats des ces programmes, mais d'abord voici ce qui fut le « déclencheur » de cette partie du rapport :

L'angle θ choisi est de 45°, et le nouveau « record » est de 66,5 mètres au lieu de 65,8…



Ce cas particulier nous indique que le mouvement du javelot est fortement influencé par ce paramètre dθ/dt. Nous constatons que la portée est plus grande, et ce pour un angle de lancer totalement différent, de même pour l'angle d'attaque. Nous pouvons en conclure que cette vitesse de rotation nous permet d'exploiter de nouvelles combinaisons de paramètres qui n'étaient pas fructueux sans elle.
Nous ne pouvons pas nous contenter d'un angle choisi au hasard et c'est pour cette raison que le dernier programme traite tous les paramètres en même temps, ces paramètres sont au nombre de trois, et de ce fait, toute représentation graphique des résultats est exclue car elle doit se faire en quatre dimensions, du moins si on veut conserver la totalité de l'information. Cependant, nous pouvons « sans problèmes » trouver les paramètres optimaux et la portée maximale.

Malheureusement, nous ne savons pas discuter de l'influence de tous les paramètres, et c'est pourquoi nous avons d'abord traité le cas où la vitesse angulaire initiale du javelot était nulle. Bien sur, le fait de lui donner une valeur ne fausse en rien notre interprétation. La vitesse angulaire est juste un paramètre supplémentaire dont nous allons étudier l'impact sur la trajectoire du javelot.
Si nous lançons le troisième programme, nous aurons déterminé tous les paramètres pour un lancer optimal, et nous serons en mesure d'observer le comportement du javelot avec les fonctions précédentes. En voici le résultat :

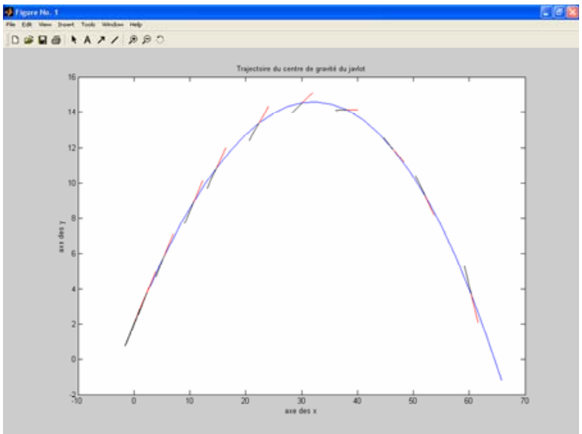


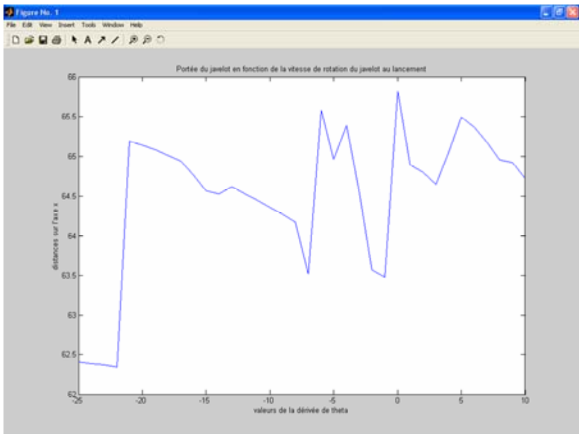
En lançant ensuite le tout premier programme, nous pouvons observer la trajectoire et la position dans l'espace du javelot.

Nous remarquons que la valeur de dθ/dt est telle que le javelot reste le plus longtemps possible dans la direction de la vitesse. A l'issue des différentes simulations que nous avions déjà faites, nous avions conclut que, pour atteindre une plus grande distance, il fallait que l'angle d'attaque soit le plus petit possible au début de la trajectoire car lorsqu'il devient trop important, il freine le javelot dans sa progression. Dans ce cas-ci, nous observons que l'angle d'attaque reste nul quasi tout le long de sa phase ascendante. Par conséquent, les forces de frottements sont moindres, nous pouvons ainsi augmenter l'angle de lancement (θ), et le javelot atteint une altitude maximale bien plus grande que s'il avait été lancé avec un dθ/dt nul. Sa course dans l'air est donc plus longue, et la portée se voit augmentée d'un peu moins d'un mètre.

Nous en concluons que la vitesse angulaire imprimée au javelot lors de son lancer joue un rôle assez important sur les performances de l'essai. Mais ce qui est surtout remarquable, c'est qu'elle influe sur tous les autres paramètres. Cependant, ce n'est pas un mystère, car tout le « secret » consiste à minimiser les forces de frottements, et ce paramètre y réussit le mieux.

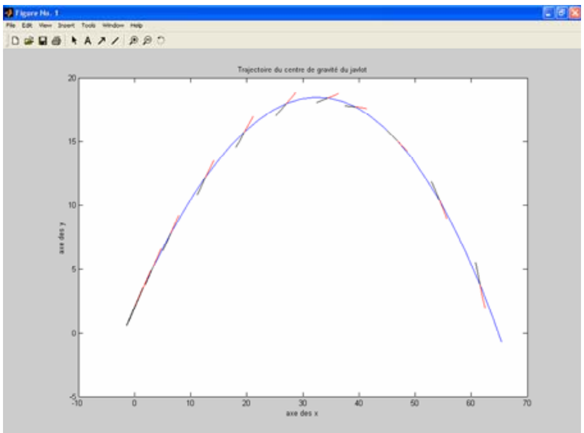
Voici quelques exemples de cas moins performants qui valident notre explication :



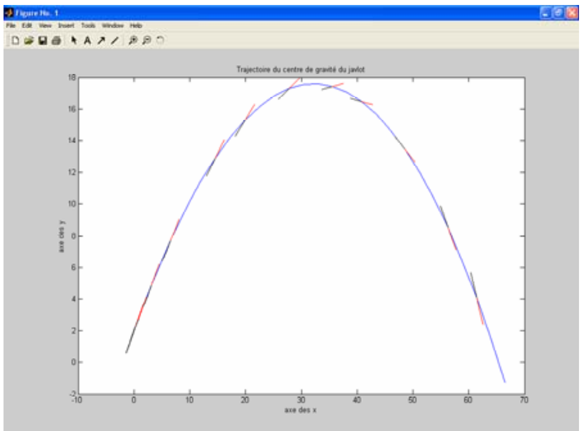


θ = 38°, α = 0°, dθ/dt = 0°/s, d = 65.82m.

Nous notons (grâce à la fonction javelotanlesDT) qu'ici dθ/dt = 0°/s maximise la portée.



θ = 45°, α = 0°, dθ/dt = 0°/s, d = 65.37m.



θ = 45°, α = 1°, dθ/dt = -10°/s, d = 66.48m.

Conclusions

A la fin de cette discussion sur le problème, nous pouvons affirmer que les paramètres qui maximisent la portée du javelot sont : $\theta = 42°$, $\alpha = 0°$, $d\theta/dt = -15°/s$, $d = 66.63m$.
Mais nous ne retiendrons pas que ces chiffres comme conclusion, nous garderons à l'esprit la manière dont chaque paramètre intervient sur la distance finale. Parfois nous pouvions prévoir ce que modifiait un paramètre, parfois pas. Prenons comme exemple $d\theta/dt$: pour un cas particulier, sa variation n'apportait aucun avantage au lancé, mais dans un autre cas, le point d'impact était plus éloigné. Cette expérience met en avant la nécessité de traiter tous les cas possibles, même si le nombre d'expériences est grand. Et pour ce faire, nous devons recourir à la puissance de calculs des ordinateurs.
De même, seule une décomposition en problèmes plus simples permet de comprendre et d'interpréter le mouvement du javelot, ainsi que l'influence qu'a chaque paramètre lors du lancer. Il est dès lors important de progresser étape par étape dans l'analyse du problème, puis de le traiter globalement. Cette méthode nous permet d'éviter certaines aberrances ou erreurs.

Enfin, nous insistons sur le fait que ces différents points ne peuvent se réaliser qu'à l'aide de logiciels sur ordinateurs, mais que de tels logiciels sont également de puissants outils pour mettre en évidence les résultats de ces programmes, et que ces résultats se doivent d'être interprétés par le programmeur afin de s'assurer qu'ils soient cohérents avec la réalité.

Annexe à la fonction « javfct.m »

Introduction

La résolution des équations du mouvement se fait par la méthode « ode45 ». Cette dernière appel une fonction qui donne une approximation numérique de la dérivée de la fonction solution, puis résout le problème en intégrant cette dérivée à différent instants successifs. Dans notre cas, la fonction dérivée est notre système d'équations adapté à une résolution de dérivée seconde.

Les équations

La fonction reçoit en entrées les valeurs des différents paramètres et de leurs dérivées. Ces valeurs sont connues (instant initial, ou déjà déterminées précédemment) et permettent d'estimer la dérivée seconde (variation de vitesse) par les trois équations données par les théorèmes généraux ou les équations de Lagrange, la dérivée première (variation de la variable) est déjà donnée en entrée de fonction. Par intégrations successives à partir d'un instant initial, ode45 fournit toutes les valeurs de chaque variable ainsi que leur vitesse.

Les forces

Ces équations font intervenir d'autres paramètres comme des angles et des forces, mais ceux-ci sont entièrement déterminés par les variables (x, y, θ) et leur dérivée première.
Le problème rencontré lors de la résolution des équations est que les formules des normes des forces généraient des nombres complexes lorsqu l'angle d'attaque devenait négatif. Nous avons donc du apporter quelques modifications à ces formules.

Les nombres complexes disparaissent définitivement si on prend la valeur absolue de l'angle d'attaque pour la force « lift ». D'autre part, il semble logique que si le vent souffle sur le dessous du javelot, la force lift s'exerce vers le haut, et si l'angle d'attaque devient négatif, le vent souffle sur le dessus, la force lift doit donc changer de sens et cela se traduit par l'introduction d'un signe moins devant la norme de cette force lorsque l'angle d'attaque devient négatif.
En ce qui concerne la force « drag », le fait que l'angle d'attaque devienne négatif se traduit par l'inverse de la norme par rapport à un angle positif. Si nous nous basons sur le bon sens, la force de traînée, ce que représente la drag force, est la même, pour un certain angle d'attaque en valeur absolue, que le vent souffle au dessus de javelot ou en dessous. Seul le fait que le vent souffle de face ou de dos pour changer le sens de la force. Afin de conserver cette logique dans notre programme, nous avons donc aussi introduit une valeur absolue sur l'angle d'attaque dans la formule de la norme de la drag force, son sens reste le même dans notre cas.

Ces quelques modifications sont les seules que nous avons apportées aux données du problèmes. Nous avons tenu à les expliquées afin de ne pas laisser le moindre doute sur nos programmes Matlab. Mais ce ne sont que des détails de modélisation qui peuvent facilement se modifier dans la fonction « javfct.m » sans avoir aucun autre impact sur tous nos autres programmes.

```
function Z = javfct(t, x)   % x = (x, x., y, y., theta, theta.)

m = 0.80625;
d = 0.255;
g = 9.81;
I = 0.42;
Z = zeros(6,1);

if x(2)~=0
    phi = atan(x(4)/x(2));
else
    if x(4)>0
        phi = pi/2;
    else
        if x(4)<0
            phi = -pi/2;
        else
            phi = 0;
        end
    end
end
alpha = x(5) - phi;

Kd = 0.00024*exp(5.157*abs(alpha));
if (alpha >= 0)
    Kl = 0.0127*(alpha^1.34);
else
    Kl = -0.0127*((-alpha)^1.34);
end
V2 = x(2)^2 + x(4)^2;

Z(1) = x(2);
Z(2) = -V2*( Kd*cos(phi) + Kl*sin(phi) )/m;
Z(3) = x(4);
Z(4) = -g -V2*( Kd*sin(phi) - Kl*cos(phi) )/m;
Z(5) = x(6);
Z(6) = -d*V2*( Kd*sin(alpha) + Kl*cos(alpha) )/I;
```

```
clear      %tapez javelot sous matlab pour le lancer

v0 = 25;   %paramètres fixes
y0 = 2;
x0 = 0;
tf = 6;    %temps final, à ajuster le point d'impact n'a pas lieu

theta0 = 45*pi/180;     %paramètres à optimiser
Dtheta0 = 0*pi/180;     %les valeurs en ° sont changées en radian
alpha0 = 0*pi/180;
phi0 = theta0 - alpha0;

[T,Z] = ode45 ('javfct', [0 tf], [x0 v0*cos(phi0) y0 v0*sin(phi0) theta0 ↙
 Dtheta0]);

Nmax = find(Z(:,3)<0);          %traitement graphique du résultat
if (Z(Nmax(1)-1,3) < -1*Z(Nmax(1),3))
    Nmax(1) = Nmax(1)-1;
end
Xmax = Z(Nmax(1),1)
X = Z(1:Nmax(1),1);
Y = Z(1:Nmax(1),3);
plot(X, Y);
title('Trajectoire du centre de gravité du javlot');
xlabel('axe des x');
ylabel('axe des y');

% hold on;      %trace le corps du javelot à différents endroits
% for N = 1:Nmax/20:Nmax
%     for i = -2:0
%         x(i+3) = Z(N,1) + i*cos(Z(N,5));
%         y(i+3) = Z(N,3) + i*sin(Z(N,5));
%     end
%     plot (x,y,'k-');
%     for i = 0:2
%         xx(i+1) = Z(N,1) + i*cos(Z(N,5));
%         yy(i+1) = Z(N,3) + i*sin(Z(N,5));
%     end
%     plot (xx,yy,'r-');
% end
% hold off;
```

```
function [DMAX, AngleMAX] = javelotangles (THETA)
                                %THETA en °, vecteur ligne
v0 = 25;
y0 = 2;
x0 = 0;
tf = 6;
Dtheta = 0*pi/180;
alpha = 0*pi/180;

D = zeros(size(THETA,2),1);
for nx = 1:size(THETA,2)
    theta = THETA(nx)*pi/180;   %en radian!
    phi = theta - alpha;
    [T,Z] = ode45 ('javfct', [0 tf], [x0 v0*cos(phi) y0 v0*sin(phi) thet↙
a Dtheta]);
    Nmax = find (Z(:,3) < 0);
    if (Z(Nmax(1)-1,3) < -Z(Nmax(1),3))
        Nmax(1) = Nmax(1)-1;
    end
    D(nx) = Z(Nmax(1),1);   %distances maximales
end
plot (THETA, D);
title('Portée du javelot en fonction de l''angle du javelot au lancement↙
');
xlabel('angles theta');
ylabel('distances sur l''axe x');

[DMAX, B] = max (D);        %distance maximale
AngleMAX = THETA(B(1));     %angle pour le réaliser
```

```
function [DMAX, AngleMAX] = javelotanglesA (alpha)
                                %alpha en °, vecteur ligne
v0 = 25;                        %theta constant, phi varie
y0 = 2;
x0 = 0;
tf = 6;
Dtheta = 0*pi/180;
theta = 45*pi/180;

D = zeros(size(alpha,2),1);
for nx = 1:size(alpha,2)
    phi = theta - alpha(nx)*pi/180;
    [T,Z] = ode45 ('javfct', [0 tf], [x0 v0*cos(phi) y0 v0*sin(phi) thet↙
a Dtheta]);
    Nmax = find (Z(:,3) < 0);
    if (Z(Nmax(1)-1,3) < -Z(Nmax(1),3))
        Nmax(1) = Nmax(1)-1;
    end
    D(nx) = Z(Nmax(1),1);   %distances max
end
plot (alpha, D);
title('Portée du javelot en fonction de l''angle d''attaque au lancement↙
 (theta cst)');
xlabel('angles alpha');
ylabel('distances sur l''axe x');

[DMAX, B] = max (D);
AngleMAX = alpha(B(1));
```

```
function [DMAX, AngleMAX] = javelotanglesAP (alpha)
                            %alpha en °, vecteur ligne
v0 = 25;                    %phi constant, theta varie
y0 = 2;
x0 = 0;
tf = 6;
Dtheta = 0*pi/180;
phi = 38*pi/180;

D = zeros(size(alpha,2),1);
for nx = 1:size(alpha,2)
    theta = phi + alpha(nx)*pi/180;
    [T,Z] = ode45 ('javfct', [0 tf], [x0 v0*cos(phi) y0 v0*sin(phi) thet↙
a Dtheta]);
    Nmax = find (Z(:,3) < 0);
    if (Z(Nmax(1)-1,3) < -Z(Nmax(1),3))
        Nmax(1) = Nmax(1)-1;
    end
    D(nx) = Z(Nmax(1),1);
end
plot (alpha, D);
title('Portée du javelot en fonction de l''angle d''attaque au lancement↙
 (phi cst)');
xlabel('angles alpha');
ylabel('distances sur l''axe x');

[DMAX, B] = max (D);
AngleMAX = alpha(B(1));
```

```
function [DMAX, ThetaMAX, AlphaMAX] = javelotanglesTotal(THETA, alpha)
                               %THETA et alpha en °, vecteur ligne
v0 = 25;
y0 = 2;
x0 = 0;
tf = 6;
Dtheta = 0*pi/180;

D = zeros(size(THETA,2),size(alpha,2));
for nx = 1:size(THETA,2)
    theta = THETA(nx)*pi/180;
    for ny = 1:size(alpha,2)
        phi = theta - alpha(ny)*pi/180;
        [T,Z] = ode45 ('javfct', [0 tf], [x0 v0*cos(phi) y0 v0*sin(phi) ↙
theta Dtheta]);
        Nmax = find (Z(:,3) < 0);
        if (Z(Nmax(1)-1,3) < -Z(Nmax(1),3))
            Nmax(1) = Nmax(1)-1;
        end
        D(nx,ny) = Z(Nmax(1),1);
    end
end
for i = 1:size(alpha,2)
    X(:,i) = THETA';
end
for i = 1:size(THETA,2)
    Y(i,:) = alpha;
end

surf(X, Y, D);
title('Portée du javelot en fonction des angles theta et alpha');
xlabel('angles theta');
ylabel('angles alpha');
zlabel('distances sur l''axe x');

[DMAX1, nT] = max (D);
[DMAX, nA] = max (DMAX1);
ThetaMAX = THETA(nT(nA));
AlphaMAX = alpha(nA);

% hold on;              %pour mettre en évidence le maximum!
% plot3(ThetaMAX, AlphaMAX, DMAX, 'g*');
% hold off;
```

```
function [DMAX, DthetaMAX] = javelotanglesDT (DTHETA)
                             %DTHETA en °, vecteur ligne
v0 = 25;
y0 = 2;
x0 = 0;
tf = 6;
theta = 38*pi/180;
alpha = 0*pi/180;
phi = theta - alpha;

D = zeros(size(DTHETA,2),1);
for nx = 1:size(DTHETA,2)
    Dtheta = DTHETA(nx)*pi/180;
    [T,Z] = ode45 ('javfct', [0 tf], [x0 v0*cos(phi) y0 v0*sin(phi) thet↙
a Dtheta]);
    Nmax = find (Z(:,3) < 0);
    if (Z(Nmax(1)-1,3) < -Z(Nmax(1),3))
        Nmax(1) = Nmax(1)-1;
    end
    D(nx) = Z(Nmax(1),1);
end
plot (DTHETA, D);
title('Portée du javelot en fonction de la vitesse de rotation du javelo↙
t au lancement');
xlabel('valeurs de la dérivée de theta');
ylabel('distances sur l''axe x');

[DMAX, B] = max (D);
DthetaMAX = DTHETA(B(1));
```

```
function [DMAX, DThetaMAX, AlphaMAX] = javelotanglesDTA(DTHETA, alpha)
                                %THETA et alpha en °, vecteur ligne
v0 = 25;
y0 = 2;
x0 = 0;
tf = 6;
theta = 40*pi/180;

D = zeros(size(DTHETA,2),size(alpha,2));
for nx = 1:size(DTHETA,2)
    Dtheta = DTHETA(nx)*pi/180;
    for ny = 1:size(alpha,2)
        phi = theta - alpha(ny)*pi/180;
        [T,Z] = ode45 ('javfct', [0 tf], [x0 v0*cos(phi) y0 v0*sin(phi) ↙
theta Dtheta]);
        Nmax = find (Z(:,3) < 0);
        if (Z(Nmax(1)-1,3) < -Z(Nmax(1),3))
            Nmax(1) = Nmax(1)-1;
        end
        D(nx,ny) = Z(Nmax(1),1);
    end
end
for i = 1:size(alpha,2)
    X(:,i) = DTHETA';
end
for i = 1:size(DTHETA,2)
    Y(i,:) = alpha;
end

surf(X, Y, D);
title('Portée du javelot en fonction de Dtheta et alpha');
xlabel('valeurs de dtheta/dt');
ylabel('angles alpha');
zlabel('distances sur l''axe x');

[DMAX1, nT] = max (D);
[DMAX, nA] = max (DMAX1);
DThetaMAX = DTHETA(nT(nA));
AlphaMAX = alpha(nA);

% hold on;              %pour mettre en évidence le maximum!
% plot3(ThetaMAX, AlphaMAX, DMAX, 'g*');
% hold off;
```

```
function [DMAX, ThetaMAX, DThetaMAX, AlphaMAX] = javelotanglesTDTA (THET↙
A, DTHETA, alpha)  %en °, vecteur ligne

v0 = 25;
y0 = 2;
x0 = 0;
tf = 6;

D = zeros(size(DTHETA,2),size(alpha,2),size(THETA,2));
for nz = 1:size(THETA,2)        %trois paramètres varient!
    theta = THETA(nz)*pi/180;
    for nx = 1:size(DTHETA,2)
        Dtheta = DTHETA(nx)*pi/180;
        for ny = 1:size(alpha,2)
            phi = theta - alpha(ny)*pi/180;
            [T,Z] = ode45 ('javfct', [0 tf], [x0 v0*cos(phi) y0 v0*sin(p↙
hi) theta Dtheta]);
            Nmax = find (Z(:,3) < 0);
            if (Z(Nmax(1)-1,3) < -Z(Nmax(1),3))
                Nmax(1) = Nmax(1)-1;
            end
            D(nx,ny,nz) = Z(Nmax(1),1);
        end
    end
    [DMAX2, nDT] = max (D(:,:,nz));       %distances maximales
    [DMAX1(nz), nA(nz)] = max (DMAX2);    %pour chaque theta
    DThetaMAX1(nz) = DTHETA(nDT(nA(nz)));
    AlphaMAX1(nz) = alpha(nA(nz));
end
[DMAX, nT] = max (DMAX1);       %La distance maximale
ThetaMAX = THETA(nT);           %avec ses paramètres adéquats
DThetaMAX = DThetaMAX1(nT);
AlphaMAX = AlphaMAX1(nT);

%trop de paramètres pour pouvoir effectuer de traitements graphiques
```