

Launch Vehicle
Simulator
1.0

Ce petit logiciel appelé « LVSIm » est un avant projet de ce qui pourra être une simulation de lancement orbital. Elle permet de définir de manière assez complète toutes les caractéristiques d'un lanceur (existant ou imaginé) et d'en tester les performances en vol.

Ce logiciel étant en cours de développement, un certain nombre d'approximations ont été faites, afin de pouvoir valider l'algorithme et la programmation.

Principales d'approximations :

- Non prise en compte du phénomène de couple et de la stabilité du lanceur.
- Non prise en compte de la variation de l'impulsion spécifique des ergols des moteurs par rapport au changement de la pression de l'air avec l'altitude.
- Non prise en compte de la variation de la direction de la poussée des moteurs lors du guidage de la fusée.
- Non prise en compte des variations des C_x et C_z par rapport à l'incidence.
- Non prise en compte de la légère variation de la gravité due à l'éloignement effectif du lanceur du centre du globe terrestre.
- Non prise en compte de la variation de poussée de propulseurs solides (du à l'augmentation de leur surface de combustion).
- Non prise en compte de la force de Coriolis.
- Non prise en compte de la mécanique orbitale en cours et en fin de mission.

Dans un futur proche, chaque élément sera pris en compte, et mieux modélisé dans la programmation du logiciel, afin de d'approcher le plus possible d'une simulation précise.

Une modélisation assez simpliste d'une fusée en vol va permettre de calculer une approximation de la composante verticale et horizontale de l'accélération subie pendant la mission, et donc de pouvoir calculer les différentiels de vitesse et de mouvement, afin de déterminer la trajectoire suivie.

Ces équations sont la base des formules qui sont utilisées dans le logiciel.

L'algorithme de « LVsim » est organisé sous forme de routines comme il suit :

- 1) Entrée de données relatives à la fusée ou utilisation du modèle « Diamant A »
- 2) Initialisation des variables induites du programme et de l'interface « Allegro ».

Début de Boucle

- 3) Réactualisation de la masse
- 4) Réactualisation des caractéristiques de l'air
- 5) Calculs cinétiques et dynamiques
- 6) Correction de trajectoire, guidage
- 7) Affichage
- 8) Enregistrement des données dans un fichier de télémétrie appelé « data.txt »
- 9) En cas de fin de vol propulsé, conclusion sur la mise en orbite

Fin de boucle

Le logiciel est programmé de manière à ce qu'une boucle représente 0.1 seconde, temps bien-sûr paramétrable.

Un certain nombre de données de base est nécessaire à la définition complète de la fusée. Elle se fait par le biais d'un fichier contenant ces données. Voici un exemple de la fusée « Diamant A » qui est définie par défaut.

```
// Stages parameters variables
// French "Diamant A" Rocket

st_nb=3;      // Number of stages

ms[1]=1946.; // Stage structure mass (kg)
ms[2]=540.;  // Stage structure mass (kg)
ms[3]=68.;   // Stage structure mass (kg)
ms[4]=10.;   // payload

mc[1]=12739.; // Stage propellant mass (kg)
mc[2]=2275.;  // Stage propellant mass (kg)
mc[3]=635.;   // Stage propellant mass (kg)

isp[1]=230.; // Stage isp (s)
isp[2]=250.; // Stage isp (s)
isp[3]=211.; // Stage isp (s)

th[1]=280000.; // Stage thrust (N)
th[2]=120000.; // Stage thrust (N)
th[3]=30000.;  // Stage thrust (N)

cx[1]=.1;      // Stage Cx
cz[1]=0.;      // Stage Cz
cx[2]=.1;      // Stage Cx
cz[2]=0.;      // Stage Cz
cx[3]=.1;      // Stage Cx
cz[3]=0.;      // Stage Cz

sa[1]=0.;      // Stage wing surface (m²)
sr[1]=1.41;    // Stage reference surface (m²)
sa[2]=0.;      // Stage wing surface (m²)
sr[2]=.50;     // Stage reference surface (m²)
sa[3]=0.;      // Stage wing surface (m²)
sr[3]=.385;    // Stage reference surface (m²)

bu_st=1 ;      // Etage en cours de combustion
```

Toutes ces données sont stockées dans des tableaux en rapport avec le type de données et vous seront demandées si vous décidez de tester votre propre fusée.

Les calculs cinétiques et dynamiques utilisent un modèle défini par M. Marty dans sa bible concernant les conceptions d'engins spatiaux.

Il s'agit d'un bilan des forces poussée, trainée, portance et pesanteur, puis d'une projection par rapport aux 2 axes du repère.

Il faut bien distinguer le « pitch » angle de la fusée par rapport à Ox et le « tangle », angle de la tangente à la trajectoire, ce qui nous permet aussi de calculer l'incidence.

```
// Calculation of the accelerations
ax=(th[bu_st]*cos(pitch*pi/180)-.5*rho*sa[bu_st]*cz[bu_st]*speed*speed*sin(tangle)-
.5*rho*sr[bu_st]*cx[bu_st]*speed*speed*cos(tangle))/m;
az=(-m*g+th[bu_st]*sin(pitch*pi/180)+.5*rho*sa[bu_st]*cz[bu_st]*speed*speed*cos(tangle)-
.5*rho*sr[bu_st]*cx[bu_st]*speed*speed*sin(tangle))/m;

// Calculation of the speeds
speedx=speedx+ax*dt;
speedz=speedz+az*dt;
speed = sqrt(speedx*speedx+speedz*speedz);

// Calculation of the coordinates
x=x+speedx*dt;
alt=alt+speedz*dt;
l=l+speed*dt;

// Angle de la tangente à la trajectoire
tangle = atan2(speedz,speedx);
```

L'affichage des données est fait par l'intermédiaire des fonctions primitives de dessin, mise à disposition par « Allegro ».

Installation et utilisation de « LVSIM » :

Il suffit de copier l'exécutable « lvsim.exe » dans un répertoire de votre choix ainsi que le fichier « alleg42.dll », qui est nécessaire au fonctionnement du programme (sauf si « Allegro » est déjà installé sur votre machine).

Il ne reste plus qu'à lancer l'application et à suivre les instructions...

Une fois la mission terminée, vous pouvez ouvrir le fichier « data.txt » qui a été créé dans ce même répertoire à l'aide du logiciel « Excel » afin de visualiser les données de vol et faire un débriefing complet.

Bons vols ...

Stéphane Query (<http://www.lvsim.online.fr>)