

Physique Numérique I – Exercice 1

à rendre jusqu'au **mercredi 12 octobre 2022** sur le site
<http://moodle.epfl.ch/mod/assign/view.php?id=835967>

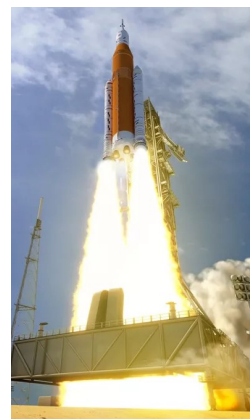
1 Lancement d'Artemis 1. Schéma d'Euler explicite

La fusée Artemis 1 a une masse au décollage de $m_0 = 2600$ tonnes. Elle comporte un ensemble de propulseurs ayant un débit massique de $D = 10000\text{kg/s}$, une vitesse d'éjection $v_e = 4200\text{m/s}$ et un temps de combustion $t_{fin} = 120\text{s}$.

En plus du poids et de la force de propulsion, la fusée subit une force de traînée aérodynamique $F_t = (1/2)\rho C_x S v^2$, avec la densité de l'air $\rho(x) = \rho_0 \exp(-x/\lambda)$, où x est l'altitude, $\rho_0 = 1.2\text{kg/m}^3$, $\lambda = 10000\text{m}$, $C_x = 0.35$ et une surface de section $S = 110\text{m}^2$.

On supposera l'accélération de la pesanteur constante $g = 9.81\text{m/s}^2$.

On utilisera le schéma numérique d'Euler explicite.



Artemis 1 au décollage
(Credit : NASA)

1.1 Calculs analytiques

- (a) [4pts] Ecrire les équations différentielles du mouvement sous la forme $\mathbf{dy}/dt = \mathbf{f}(\mathbf{y})$, où \mathbf{y} est le vecteur (x, v, m) et \mathbf{f} est un vecteur de fonctions à déterminer.
- (b) [4pts] Trouver la solutions analytique dans le cas $C_x = 0$.
- (c) [4pts] Ecrire les expressions de l'énergie mécanique de la fusée, E_{mec} , et de la puissance des forces non conservatives, P_{nc} . Ecrire le théorème de l'énergie mécanique pour la fusée. (*Indication : multiplier l'équation différentielle pour dv/dt par v et faire attention au fait que la masse de la fusée dépend du temps*).

1.2 Implémentation en C++

Téléchargez le squelette du code C++ (.cpp, .hpp et .h), un exemple de fichier d'input (configuration.in.example), une ébauche de script Matlab (.m) et des fonctions Python (.py) ici : [Exercice1.2022.zip](#). Renommez fichier source C++ (.cpp) en `Exercice1_VotreNom.cpp`. Vous complétez le code C++ en programmant l'algorithme d'Euler explicite et en ajoutant le calcul de l'énergie mécanique et de la puissance des forces non conservatives.

Compiler avec la commande : `g++ -o Exercice1.exe Exercice1_VotreNom.cpp` ou en utilisant le `Makefile` donnée avec la commande `make student` depuis un terminal Linux.

Exécuter avec la commande : `./Exercice1.exe configuration.in.example`, après avoir modifié selon vos besoins le fichier d'input `configuration.in.example`.

Attention : tout fichier préexistant ayant le même nom que la variable output du fichier d'input sera écrasé.

Analyser les résultats et faire des figures. On utilisera, au choix, Matlab ou python. Pour Matlab, on donne une ébauche de script, `Analyse_Ex1_student.m`, que vous devrez modifier pour l'adapter à vos besoins. Avec python3, on donne une ébauche de fonction, `Analyse_Ex1_student.py`. Il suffit de 1) configurer le fichier `Analyse_Ex1_student.py` avec les inputs de votre choix et 2) exécuter la commande : `python3 Analyse_Ex1_student.py` dans le terminal Linux. Ne pas oublier que l'utilisation de `Analyse_Ex1_student.py` nécessite d'avoir installé les librairies suivantes : `numpy`, `matplotlib`, `scipy` et `os`. Des indications pour leur installation sont contenues dans le header du fichier `Analyse_Ex1_student.py`.

1.3 Simulations et Analyses

- (a) [10pts] **Cas** $C_x = 0$. Effectuer une série de simulations en faisant varier le nombre de pas de temps (par exemple $N = 5, 10, 20, 40, 80, 160, \dots$). Vérifier la convergence des résultats numériques vers la solution analytique lorsque Δt tend vers zéro. Quelle est la dépendance en Δt de l'erreur (i.e. l'ordre de convergence) sur la vitesse finale et l'altitude finale ?
- (b) [6pts] On considère maintenant l'effet de la force de traînée avec $C_x = 0.35$. Effectuer une série de simulations avec les nombres de pas de temps différents et étudier la convergence numérique de la position et de la vitesse finales.
- (c) [7pts] Vérifier le théorème de l'énergie mécanique : $dE_{\text{mec}}/dt = P_{\text{nc}}$. Pour évaluer la dérivée temporelle, on utilisera les différences finies centrées, $dE_{\text{mec}}(t_j)/dt \approx (E_{\text{mec}}(t_{j+1}) - E_{\text{mec}}(t_{j-1})) / (2\Delta t)$.
- (d) [10pts] Effectuer des simulations pour un temps final plus long, $t_{\text{fin}} = 260\text{s}$. Faire une étude de convergence de la position et de la vitesse finales, dans les cas $C_x = 0$ et $C_x = 0.35$.

1.4 Rédaction du rapport en L^AT_EX

- (a) Télécharger le fichier source L^AT_EX ([SqueletteRapport.tex](#)) du site "Moodle" qui peut servir comme base de départ.
- (b) Rédiger un rapport dans lequel les résultats des simulations ainsi que les réponses aux questions ci-dessus sont présentés et discutés.

N.B. On trouve plusieurs documents L^AT_EX (introduction, exemples, références) dans un dossier spécifique sur notre site "Moodle" ([Dossier L^AT_EX](#)).

1.5 Soumission du rapport en format pdf, de la source L^AT_EX et du fichier source C++

- (a) Préparer le fichier du rapport en format pdf portant le nom : **RapportExercice1_VotreNom.pdf** et la source L^AT_EX : **RapportExercice1_VotreNom.tex**
 - (b) Préparer votre fichier source C++ sous le nom **Exercice1_VotreNom.cpp**.
 - (c) Préparer votre fichier script Matlab ou python3 sous le nom **Analyse_Exercice1_VotreNom.m** ou **Analyse_Exercice1_VotreNom.py**.
 - (d) Le formulaire de soumission se trouve dans la case du premier exercice sur notre site Moodle et peut directement être atteint en cliquant [ici](#).
- [5pts] En plus des points mentionnés ci-dessus, **5 points sont attribués pour la qualité générale de la présentation du rapport (clarté des arguments, des figures, etc.)**

Note = 10 + total des points obtenus