Consignes :

Vous avez le droit à vos documents de cours (une correction des derniers TP est disponible ici : https://www.cpge-brizeux.fr/wordpress/category/pcsi/info-pcsi-1920)

Nous attendons un compte rendu en deux parties : votre programme python et un fichier texte (pouvant contenir des photos). Taille maximale de l'ensemble : 2 Mo. Le programme Python contiendra les réponses aux questions où il est explicitement demandé de programmer (n'oubliez pas de commenter votre code). Le fichier texte contiendra les réponses aux autres questions (Q3, Q4, Q6, Q8, Q11, Q14). Encore une fois, vous pouvez faire des captures d'écran et insérer des photos.

Le TP dure 2h. Vous avez ensuite jusqu'à lundi 12h pour déposer vos fichiers sur le cloud à l'adresse : https://cloud.cpge-brizeux.fr/index.php/s/ke33yg9dstREXDC

TP Résolution d'équation et analyse de fonction

1. Méthode d'Euler

Durant les derniers TPs, nous avions résolu analytiquement ainsi qu'avec la méthode d'Euler l'équation différentielle suivante :

$$\begin{cases} y' = y^2 \\ y(0) = \dfrac{1}{2} \end{cases} \qquad (1)$$

Cette équation différentielle est un sous cas de l'équation différentielle (2) :

$$\begin{cases} y' = y^N \\ y(0) = y_0 \end{cases} \qquad (2)$$

Dans cette partie du TP, nous étudierons l'équation (2) **sur l'intervalle** $[\mathbf{0}\,;\mathbf{1}]$.

Q1 : Programmer une fonction `Euler_n(n,N,y0)` qui prend trois paramètres n, N et y_0 ; résout l'équation (2) sur l'intervalle $[0\,;1]$ avec la méthode d'Euler pour n points et renvoie la liste des y obtenus grâce à l'approximation.

Lors du précédent TP sur la méthode d'Euler, nous avions résolu (1) et obtenu la solution (3) :

$$y(t) = \frac{1}{2-t} \qquad (3)$$

Q2 : Programmer une fonction `Comparaison(n)` qui affiche sur un même graphe la solution exacte (3) ainsi que l'approximation obtenue avec la fonction `Euler_n(n,N,y0)` pour n points.

Tout comme l'équation différentielle (1), l'équation différentielle (2) possède une solution exacte de la forme (4) :

$$y(t) = ({y_0}^{1-N} + (1-N).t)^{\frac{1}{1-N}} \qquad (4)$$

Q3 : Démontrer que l'équation (4) est solution de l'équation différentielle (2).

Q4 : Vérifier que la solution (3) est bien un sous cas de la solution (4).

Q5 : Programmer une fonction `Comparaison2(n,N,y0)` qui affiche sur un même graphe la solution exacte (4) ainsi que l'approximation obtenue avec la fonction `Euler_n(n,N,y0)` pour n points.

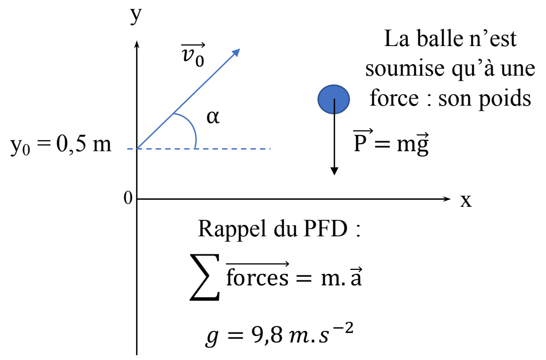
Q6 : Insérer dans votre compte rendu le résultat obtenu avec la fonction `Comparaison2` résolvant l'équation (5) pour $n = 50$.

$$\begin{cases} y' = y^3 \\ y(0) = 0{,}01 \end{cases} \qquad (5)$$

Q7 : Programmer une fonction `Comparaison3(n,y0)` qui affiche pour N allant de 2 à 7 par pas de 1, sur un même graphe, les solutions exactes de (4) ainsi que les approximations obtenue avec la fonction `Euler_n(n,N,y0)` pour n points.

2. Dichotomie et Newton

Dans le cadre d'un TP de mécanique, un élève lance une petite balle vers le haut avec une vitesse initiale $v_0 = 5\ m.s^{-1}$ et un angle α par rapport à l'axe des abscisses. Il lance sa balle depuis une hauteur $y_0 = 0.5\ m$ au dessus de la paillasse de TP.



Grâce aux équations de la mécanique, il trouve que l'évolution de la hauteur de la balle en fonction du temps vaut :

$$y(t) = -\frac{1}{2}g.t^2 + v_0.sin\alpha.t + y_0 \qquad (6)$$

La référence d'altitude (y = 0m) se situe au niveau de la paillasse de TP. L'élève souhaite savoir à quel moment la balle passe en dessous de la paillasse de TP (altitude négative) et connaître le moment où elle atteint son attitude maximale.

On étudiera le mouvement sur l'intervalle **[0 ; 1.5] secondes** avec une vitesse initiale de 5 m.s^{-1}.

Q8 : En utilisant le PFD, retrouvez l'équation (6).

Q9 : Programmer une fonction qui trace n points de la courbe pour un angle de 45°.

On utilisera toujours un angle de 45° dans les questions suivantes.

Q10 : Grâce à la méthode de Newton, programmer une fonction `temps_chute` qui renvoie une valeur approchée (avec une précision epsilon) du temps au bout duquel la balle repasse au niveau de la paillasse (altitude = 0 m).

Q11 : À la main, trouvez le temps pour lequel l'altitude de la balle est maximale (sommet de la parabole).

Q12 : Grâce à la méthode de la dichotomie, programmer une fonction `dichotomie_maximum` qui renvoie une valeur approchée (avec une précision epsilon) du temps pour lequel l'altitude de la balle est maximale.

Q13 : Grâce à la méthode de Newton, programmer une fonction `Newton_maximum` qui renvoie une valeur approchée (avec une précision epsilon) du temps pour lequel l'altitude de la balle est maximale.

À présent, l'élève veut prévoir l'instant auquel la balle touchera le sol (altitude = -0.80 m).

Q14 : Proposer un nouveau critère pour choisir dans quelle moitié de l'intervalle de départ se trouve la valeur de temps telle que l'altitude vaut -0,80m.

Q15 : Grâce à la méthode de la dichotomie, programmer une fonction `dichotomie_sol` qui renvoie une valeur approchée (avec une précision epsilon) du temps pour lequel la balle touche le sol.