Université de Montpellier **Année Universitaire 2024-2025**
Faculté des Sciences
Mathématiques

TP noté HAX604X.

Déposer vos fichiers (2 au maximum) sur l'espace moodle de l'UE HAX604X, dans l'espace de dépôt TP noté. Pas d'envoi par mail ! *Vous pouvez déposer des fichiers* ***.ipynb****. ou* ***.py*** *(Python).* La notation tiendra compte de la lisibilité du code, des commentaires et des légendes des figures. Les deux exercices sont indépendants. Le barème est indicatif.

1 Codage du schéma des 3/8 de Newton (14 pts).

Le schéma des 3/8 de Newton pour résoudre l'équation différentielle $y' = f(t, y)$ est défini par :

$$k_1 = f(t_0, y_0)$$
$$k_2 = f(t_0 + \frac{h}{3}, y_0 + \frac{h}{3}\, k_1)$$
$$k_3 = f\left(t_0 + \frac{2h}{3}, y_0 - \frac{h}{3}\, k_1 + h\, k_2\right)$$
$$k_4 = f\left(t_0 + h, y_0 + h(k_1 - k_2 + k_3)\right)$$
$$y_1 = y_0 + \frac{h}{8}\, (k_1 + 3k_2 + 3k_3 + k_4)$$

Codez ce schéma dans une fonction `N38(f, t0, y0, h, tfin)`. Puis en prenant la fonction $f(t, y) = y$ et les paramètres : intervalle de temps `t0`, `tfinal` $= 0, 7$, donnée initiale `y0` $= 1$, exécutez les tâches suivantes.

- tracer sur la même figure pour le pas de temps $h = 1$ la solution numérique calculée par le schéma ainsi que la solution exacte.
- calculer les solutions approchées par le schéma N38 pour une suite de pas de temps $h = 1/8, 1/16, 1/32, 1/64, 1/128$.
- tracer sur une autre figure en coordonnées logarithmiques l'erreur max entre la solution exacte et la solution numérique en fonction du pas h.

D'après vos résultats, quel est *l'ordre de convergence de ce schéma* ? Justifiez en traçant une droite de pente convenable dans le graphe d'erreur en coordonnées log-log.

2 Codage de l'orbite d'Arenstorf (6 pts).

On considère le système d'équations différentielles :

$$x'' = x + 2y' - \mu_2\, \frac{(x + \mu_1)}{D_1} - \mu_1\, \frac{(x - \mu_2)}{D_2}$$
$$y'' = y - 2x' - \mu_2\, \frac{y}{D_1} - \mu_1\, \frac{y}{D_2}$$

où on a noté les dénominateurs

$$D_1(x, y) = \left((x + \mu_1)^2 + y^2\right)^{3/2}$$
$$D_2(x, y) = \left((x - \mu_2)^2 + y^2\right)^{3/2}$$

Les valeurs des paramètres sont $\mu_1 = 0.012277471$ et $\mu_2 = 1 - \mu_1$.

1. Ecrire le système sous la forme d'une équation différentielle du premier ordre $Z' = F(t, Z)$ où $Z = (x, y, x', y')^T$ est un vecteur à 4 composantes. Préciser la fonction F.
2. *En utilisant le solveur* solve_ivp de Python, calculer la solution numérique sur l'intervalle de temps $[0, 17]$ avec les conditions initiales suivantes (attention aux valeurs numériques elles sont précises) :

   $$x(0) = 0.994,\ y(0) = 0,\ x'(0) = 0,\ y'(0) = -2.001585.$$

   ```
   from scipy.integrate import solve_ivp
   sol=solve_ivp(F, [t0, tfinal], Z0,  t_eval=np.linspace(t0, tfinal, 1000), rtol=1e-6, atol=1e-6)
   ```

   Les valeurs du vecteur calculées par le solveur sont dans le tableau `sol.y`, (pour voir toutes les options, consulter l'aide `https://docs.scipy.org/doc/scipy/`.)
3. Tracer la trajectoire $t \mapsto (x(t), y(t))$ pour `t` $=$ `np.linspace(0, tfinal, 1000)`. Prenez d'abord `tfinal` $= 17$.
   La trajectoire a été calculée par R. Arenstorf pour la mission Apollo 11. Elle représente l'orbite de la capsule soumise à l'attraction gravitationnelle de la terre et de la lune. La trajectoire est *périodique. Estimez approximativement la période*, donnez une décimale. Modifiez légèrement $x(0) = 0.995$. Que se passe-t-il ?