Runge Kutta

1) Ecrire une fonction $f(y,t)$ qui retourne la valeur de :

$$y - t^2 - 1$$

2) Ecrire une fonction $RK4(f, y_0, t_0, t_1, h)$ qui calcule la valeur de

$$y_{i+1} = y_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

$$t_i = t_0 + i \cdot h$$

dans l'intervalle t_0, t_1, avec :

$$y_0 = y_0$$

$$k_1 = h \cdot f(y_i, t_i)$$

$$k_2 = h \cdot f\left(y_i + \frac{k_1}{2}, t_i + \frac{h}{2}\right)$$

$$k_3 = h \cdot f\left(y_i + \frac{k_2}{2}, t_i + \frac{h}{2}\right)$$

$$k_4 = h \cdot f(y_i + k_3, t_i + h)$$

3) Ecrire une fonction $f2(t)$ qui calcule la valeur de

$$t^2 + 2t + 1 + \frac{1}{2} e^t$$

4) Tracer la valeur de $Y = RK4(f, y_0, t_0, t_1, h)$ en fonction de t pour

$y_0 = 0.5$

$t_0 = 0$ $\quad h \in \{1.0; 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3\}$

$t_1 = 4$

et la valeur de $f2(t)$ et enregistrer le résultat dans le fichier "RK4.py".

La fonction RK4 implement l'algorithme de Runge-Kutta de 4ème ordre. Cette algorithme intègre numériquement les equation differentielle:

$$y' = f(y, t)$$

avec la condition initial $y(t_0) = y_0$.
C'est l'un des algorithmes les plus utilisés pour intégrer les équations différentielles et ça precision varie directement avec la taille, h, du pas. Versions plus sophistiquées varient automatiquement la valeur de h à chaque étape pour minimiser l'erreur.