Université de Montpellier **Année Universitaire 2023-2024**
Faculté des Sciences
Mathématiques
L3 : module HAX604X Analyse numérique des équations différentielles.

1 Le schéma des trapèzes implicites.

Le schéma des trapèzes *implicite* (appelé aussi schéma de Crank-Nicolson) est défini par :

$$y_1 = y_0 + \frac{h}{2} \left\{ f(t_0, y_0) + f(t_0 + h, y_1) \right\}$$

Ainsi y_1 est solution d'un *problème de point fixe.* Quand h est petit et que $y \mapsto f(t, y)$ est lipschitzienne, on montre facilement que $g : z \mapsto y_0 + \frac{h}{2} \{f(t_0, y_0) + f(t_0 + h, z)\}$ est contractante et donc qu'il y a un unique point fixe y_1. Pour calculer effectivement y_1, on effectue quelques itérations de point fixe $y_1^{(k+1)} = g(y_1^{(k)})$ en initialisant par $y_1^{(0)} = y_0 + hf(t_0, y_0)$. En pratique on effectue trois itérations, sans effectuer de test d'arrêt. Une fois calculé y_1 on avance au pas de temps suivant : $t = t + h$ et on recommence.

Dans tout ce qui suit, on s'intéresse à l'équation différentielle :

$$y' = 3y - 4\exp(-t) \tag{1}$$

dont la solution générale est $\exp(-t) + C \cdot \exp(3t),\ C \in \mathbb{R}$.
Si on prescrit la *condition initiale* $y(0) = 1$ la **solution exacte** est $y(t) = \exp(-t)$.

2 Codage du schéma des trapèzes implicites. (TP noté 2019).

1. Vous écrirez un code qui réalisera les tâches suivantes :
   — calcule les solutions approchées de l'équation différentielle (1) avec la condition initiale $y(0) = 1$ par le schéma des trapèzes implicite pour une suite de pas de temps de plus en plus petits : $h, h/2, h/4 \ldots, h/512$.
   — trace sur la même figure pour le pas de temps final $h/512$ la solution numérique calculée ainsi que la solution exacte $t \mapsto \exp(-t)$.
   — affiche sur la console l'erreur $\max_n(|y_n - \exp(-t_n)|)$ pour le pas de temps le plus petit $h/512$.
   Exécutez ce code avec les paramètres : intervalle de temps $[\mathtt{t0}, \mathtt{tfinal}] = [0, 4]$, pas de temps initial $\mathtt{h} = 1$, donnée initiale $\mathtt{y0} = 1$.
2. Complétez votre code afin qu'il trace sur une autre figure en coordonnées logarithmiques `loglog` l'erreur $e(h) := \max_n(|y_n - \exp(-t_n)|)$ entre la solution exacte et la solutions numérique en fonction du pas h.
   D'après vos résultats, quel est l'ordre de convergence de la méthode des trapèzes implicites ? Justifiez numériquement votre réponse, par exemple en affichant sur la même figure une droite de pente convenable.
3. L'algorithme standard de MATLAB ou Python pour résoudre les équations différentielles est le schéma de Dormand-Prince d'ordre 5 à pas adaptatif avec estimation d'erreur d'ordre 4), désigné par l'acronyme `RK45`. C'est un schéma à un pas explicite de type Runge-Kutta. Faire tracer sur la même figure la solution obtenue par `RK45`, la solution obtenue par votre code des trapèzes implicites et la solution exacte. Que pensez vous des résultats ? Que se passe-t-il si on augmente légèrement `tfinal` ? Avez vous une explication ?

Consignes : Les questions traitées seront séparées par des sections de votre code. Vous fournirez les réponses aux questions dans des commentaires.

La commande de résolution de base pour résoudre avec le solveur `RK45` sont
en Matlab :

```
f=@(t,y) 3*y - 4*exp(-t);
ode45(f,[t0, tfinal], y0]
```

en Python :

```
from math import *
f = lambda t,y : 3*y - 4*exp(-t)
from scipy.integrate import solve_ivp
solve_ivp(f, [t0, tfinal], [y0], 'RK45')
```

Pour voir toutes les options et possibilités consulter l'aide de Matlab ou `Scipy`.