Université Montpellier
Faculté des Sciences Montpellier
Département de Mathématiques et Informatique

Module : Estimation à Posteriori

Master 2 : Modélisation et Analyse Numérique (MANU)

Rapport du séance 1

Réalisé par : NAOUAL Arkhouch

Année universitaire
2024/2025

Table des matières

Introduction

Le but de ce projet consiste à comparer la solution exacte et numérique d'une équation différentielle (1) en utilisant la méthode d'Euler explicite.

L'équation aux dérivées partielles (EDP) donnée par

$$\begin{cases} u'(t) = \lambda u(t) \\ u(0) = 1 \end{cases} \tag{1}$$

avec avec λ est une constante.

0.1 Solution exacte

Trouvons la solution exacte d'équation différentielle (3).
On a :

$$u'(t) = \lambda u(t) \tag{2}$$

Alors l'équation est :

$$\frac{du}{dt} = -\lambda u \quad u(0) = 1$$

Intégration des deux côtés :
En supposant que la fonction u est à valeurs strictement positives sur l'intervalle I, on trouve alors :

$$ln(u) = -\lambda t + c$$

En exponentiant des deux côtés :
Alors

$$u(t) = e^{-\lambda t + c}$$

Utilisation de la condition initiale $u(0) = 1$ pour déterminer e^c

$u(0) = e^c = 1$ alors c=0

Donc, la solution exacte est :

$$u(t) = e^{-\lambda t}$$

Si $\lambda = 1$ alors

$$u(t) = e^{-t}$$

0.2 Méthode numérique

Dans cette partie on va présenter deux méthodes numériques pour résoudre notre problème (3) ; la méthode d'Euler explicite On a :

$$\begin{cases} u'(t) = \lambda u(t) \\ u(0) = 1 \end{cases} \tag{3}$$

avec $\lambda = 1$ Étudions le cas du schéma d'Euler explicite. On obtient :

$$\frac{u_{n+1} - u_n}{dt} = -\lambda u_n$$

donc :

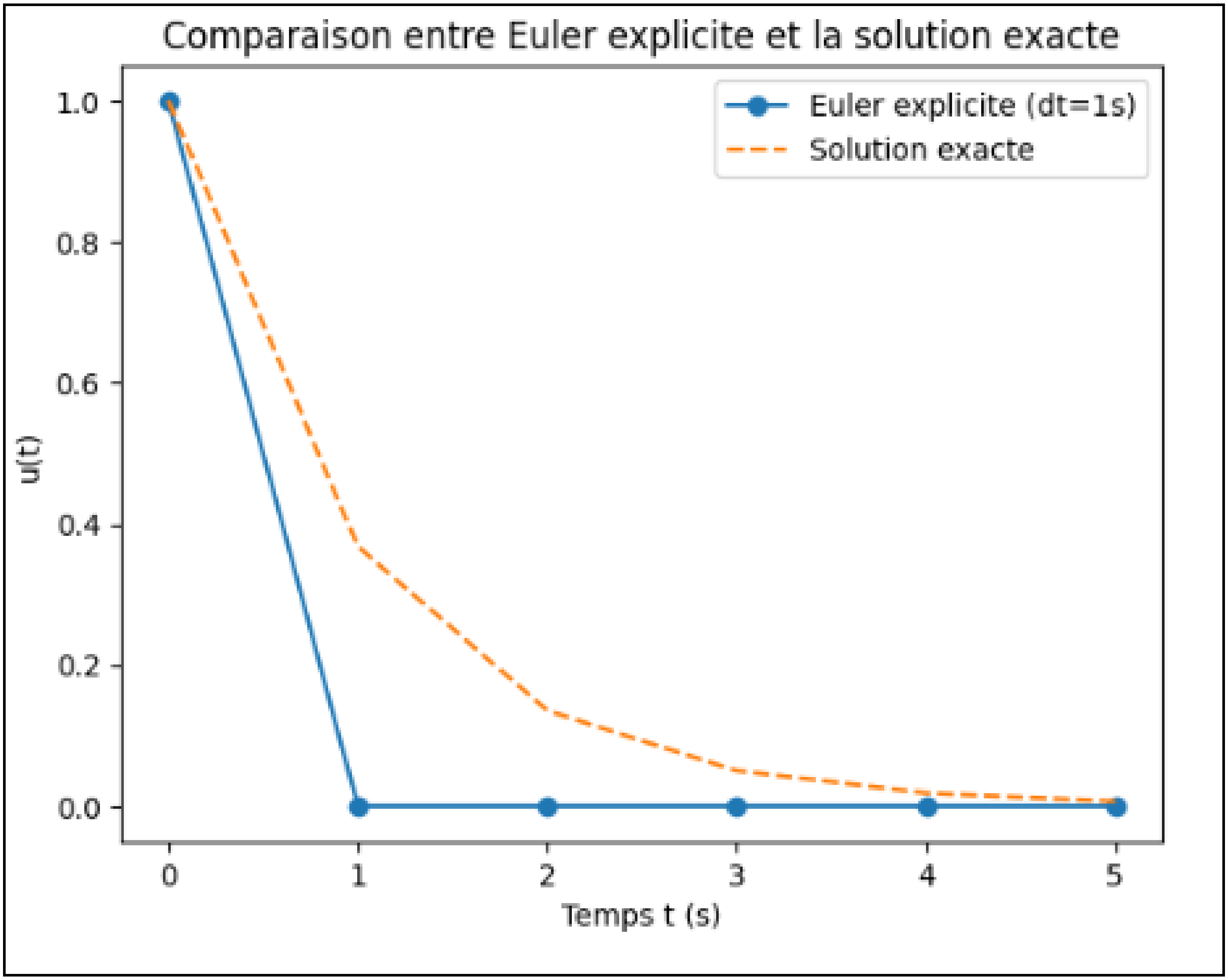
$$u_{n+1} = u_n - dt\lambda u_n \tag{4}$$

avec $u_0 = 1$.

Pour la figure ci-dessous, le codage utilisé pour tracer la solution exacte et la solution numérique u en fonction Dt = 1s.

```
import numpy as np
import matplotlib.pyplot as plt

# Paramètres
lambda_ = 1       # Valeur de lambda
u0 = 1            # Condition initiale
t_final = 5       # Temps final
dt = 1            # Pas de temps (Dt = 1s)
N = int(t_final / dt)  # Nombre d'itérations

# Initialisation des tableaux
t = np.linspace(0, t_final, N+1)  # Tableau des instants de temps
u_euler = np.zeros(N+1)           # Tableau des solutions numériques
u_euler[0] = u0                   # Valeur initiale

# Boucle d'Euler explicite
for n in range(N):
    u_euler[n+1] = u_euler[n] - lambda_ * dt * u_euler[n]

# Solution exacte pour comparaison
u_exact = u0 * np.exp(-lambda_ * t)

# Tracé des résultats
plt.plot(t, u_euler, label="Euler explicite (dt=1s)", marker='o')
plt.plot(t, u_exact, label="Solution exacte", linestyle='dashed')
plt.xlabel('Temps t (s)')
plt.ylabel('u(t)')
plt.legend()
plt.title("Comparaison entre Euler explicite et la solution exacte")
plt.show()
```

Figure 2 : La solution exacte et La solution numérique en fonction de temps t en utilisant méthode d'Euler explicite



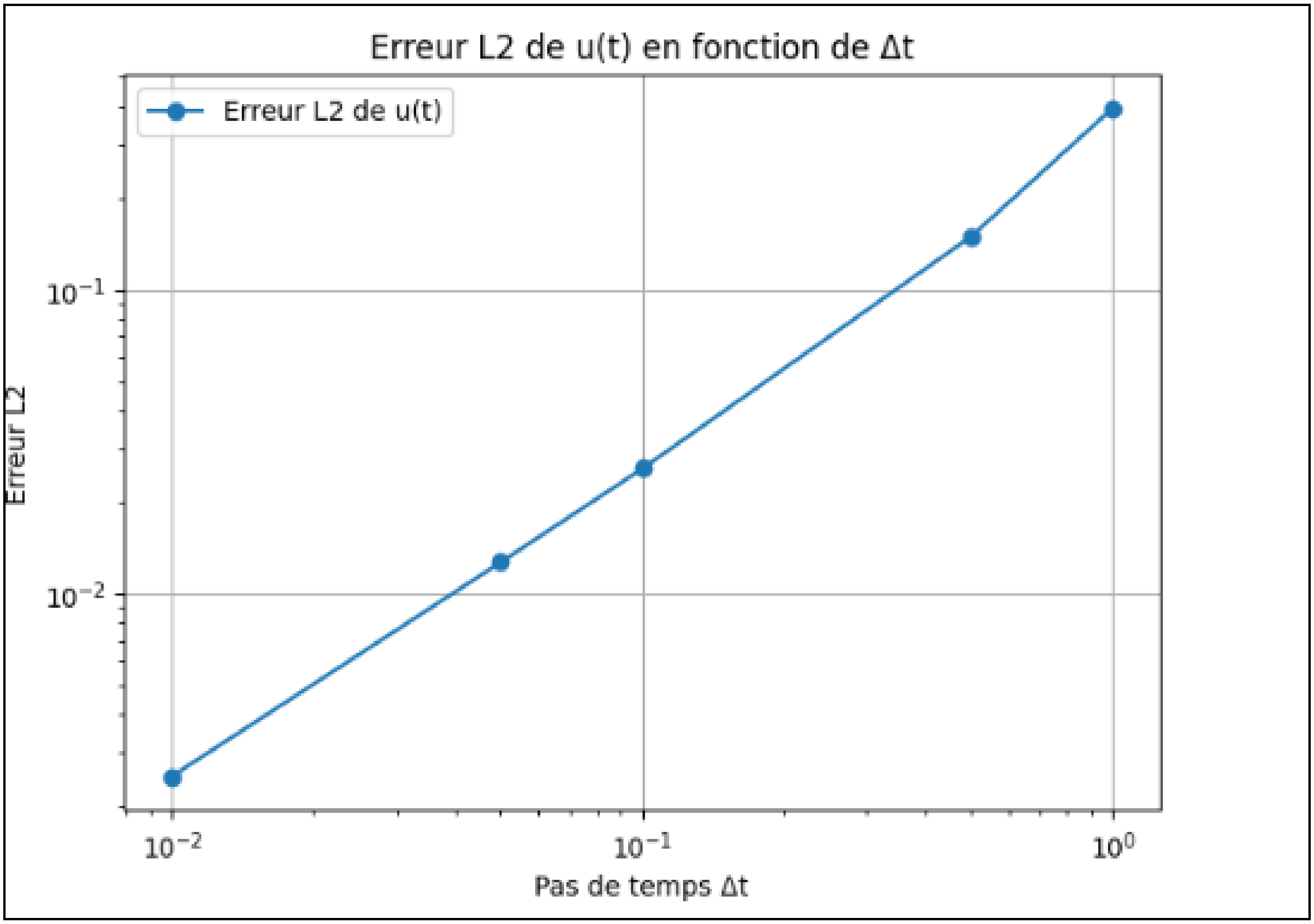
0.3 L'erreur L^2 entre la solution exacte et numérique

Dans cette partie, nous allons tracer la variable L^2

Pour la figure ci-dessous, le codage utilisé pour tracer L'erreur L^2 entre la solution exacte et numérique.

```python
import numpy as np
import matplotlib.pyplot as plt

# Paramètres
lambda_ = 1         # Valeur de lambda
u0 = 1              # Condition initiale
t_final = 60        # Temps final (1 minute)
dt_values = [1, 0.5, 0.1, 0.05, 0.01]  # Différents pas de temps à tester

# Liste pour stocker les erreurs L2 de u(t)
erreurs_L2_u = []

# Boucle sur les différents pas de temps
for dt in dt_values:
    N = int(t_final / dt)  # Nombre d'itérations
    t = np.linspace(0, t_final, N+1)  # Tableau des instants de temps
    u_euler = np.zeros(N+1)           # Tableau des solutions numériques
    u_euler[0] = u0                   # Valeur initiale

    # Boucle d'Euler explicite
    for n in range(N):
        u_euler[n+1] = u_euler[n] - lambda_ * dt * u_euler[n]

    # Solution exacte pour comparaison
    u_exact = u0 * np.exp(-lambda_ * t)

    # Calcul de l'erreur L2 pour la fonction u(t)
    erreur_L2_u = np.sqrt(np.sum(dt * (u_exact - u_euler)**2))
    erreurs_L2_u.append(erreur_L2_u)

# Tracé de l'erreur L2 de u(t) en fonction du pas de temps
plt.figure(figsize=(7, 5))
plt.plot(dt_values, erreurs_L2_u, marker='o', label="Erreur L2 de u(t)")
plt.xscale('log')
plt.yscale('log')
plt.xlabel('Pas de temps Δt')
plt.ylabel('Erreur L2')
plt.title("Erreur L2 de u(t) en fonction de Δt")
plt.grid(True)
plt.legend()
```

Figure 4 : L'erreur L^2 entre la solution exacte et numérique



Comme l'erreur diminue lorsquele tempt diminue, cela confirme que la méthode d'Euler explicite est convergente pour ce problème.