TD 1

Tom Wozniak

11 septembre 2025

1 Introduction

La première séance a consisté à prendre en main le langage de programmation Python, en résolvant numériquement quelques équations différentielles aux dérivées partielles (EDP) classiques. Nous avons utilisé des intelligences artificielles génératives pour générer les programmes.

2 Premiers exemples

2.1 L'équation $u'(t) = -u(t)$

Nous avons commencé par étudier une équation différentielle extrêmement classique. Le but de l'exercice était de résoudre numériquement avec un schéma d'Euler explicite l'équation

$$u'(t) = -u(t), \quad u(0) = 1.$$

La solution de ce problème de Cauchy est la fonction $t \mapsto \exp(-t)$, et nous avons comparé la solution exacte avec la solution approchée (voir figure 1) pour un pas de temps $\Delta t = 1$.

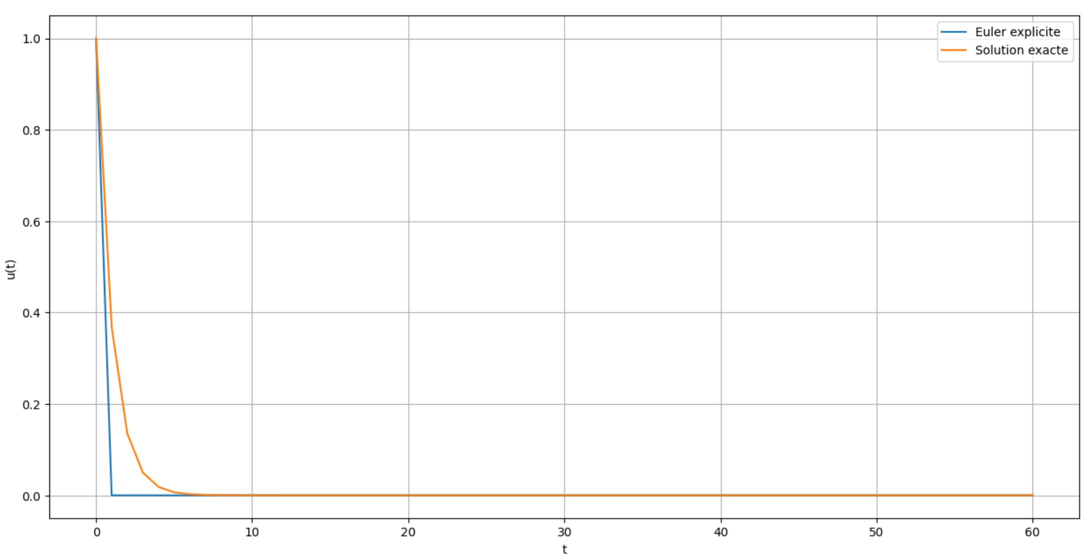

Figure 1 – Graphe comparant la solution exacte et la solution approchée de l'équation pour un pas de temps de 1.

Nous avons ensuite écrit un code permettant, via une boucle imbriquée, de créer un graphe avec échelle logarithmique, montrant l'erreur quadratique de la solution approchée en fonction du pas de temps, ce dernier variant de 1 à 0.001, comme le montre la figure 2.

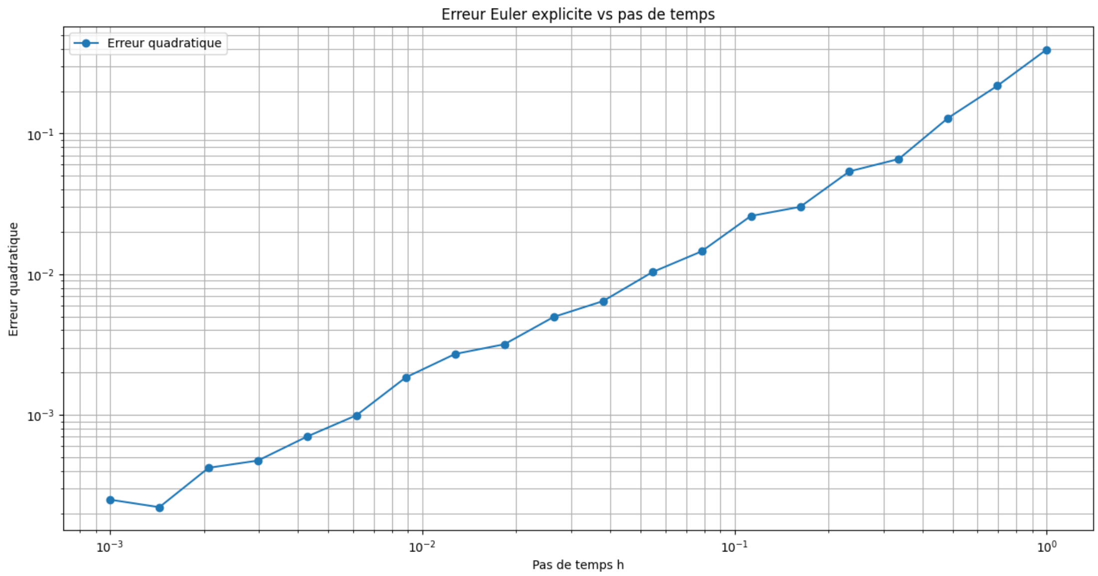

FIGURE 2 – Graphe de l'erreur quadratique de la solution approchée en fonction du pas de temps.

2.2 L'équation d'advection-diffusion en une dimension

Dans un second temps, nous nous sommes intéressés à l'équation d'advection-diffusion en une dimension. L'équation considérée est

$$\begin{cases} u_t + v_1\, u_x - \nu\, u_{xx} = -\lambda u + f, & t > 0,\ 0 < x < L, \\ u(0,x) = u_0(x), & 0 \le x \le L, \\ u(t,0) = 1, \quad u_x(t,L) = 0, & t \ge 0. \end{cases} \tag{1}$$

Le but était de trouver une fonction u_0 convenable vérifiant les conditions aux limites. Par exemple, la fonction $u_0(x) = \exp(-10x^2)$ convient.

2.3 Une dernière équation

Enfin, nous nous sommes intéressés à l'équation

$$f(t,s) = T_c \exp\Big(-k\, d(s,s_c)^2\Big), \quad d(s,s_c)^2 = (s_1 - s_{c,1})^2 + (s_2 - s_{c,2})^2 \tag{2}$$

Le but était alors de résoudre cette équation numériquement puis d'illustrer la solution, l'erreur quadratique et la norme de son gradient sur trois figures côte à côte.

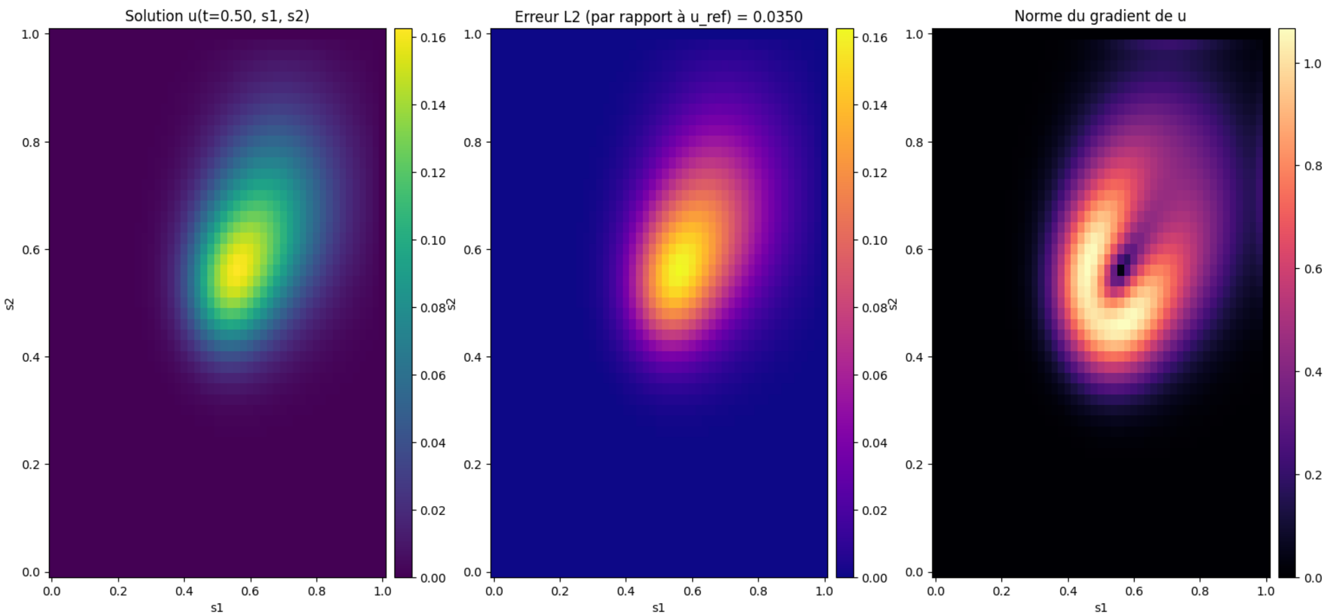

Figure 3 – Graphes montrant la solution, l'erreur quadratique et la norme du gradient en fonction de s_1 et s_2