Portier Adrien

LMAT1151 - Calcul numérique : méthodes et outils logiciels
TP6 : Intégration numérique

Pour analyser l'erreur entre la vraie valeur de l'intégrale, et celle donnée par Scipy, nous proposons d'analyser comme proposé dans l'énoncé, dans un intervalle entre 0 et 1. De plus, il serait intéressant d'avoir un graphe à analyser pour voir les fluctuations, dès lors, posons la fonction définie pour tout $t \in A$

$$F(t) = \int f(t)\,\mathrm{d}t$$

où $f : A \to \mathbb{R}$ sera la fonction à intégrer avec $A \subset \mathbb{R}$ et où on regardera généralement l'intégrale définie entre deux bornes.

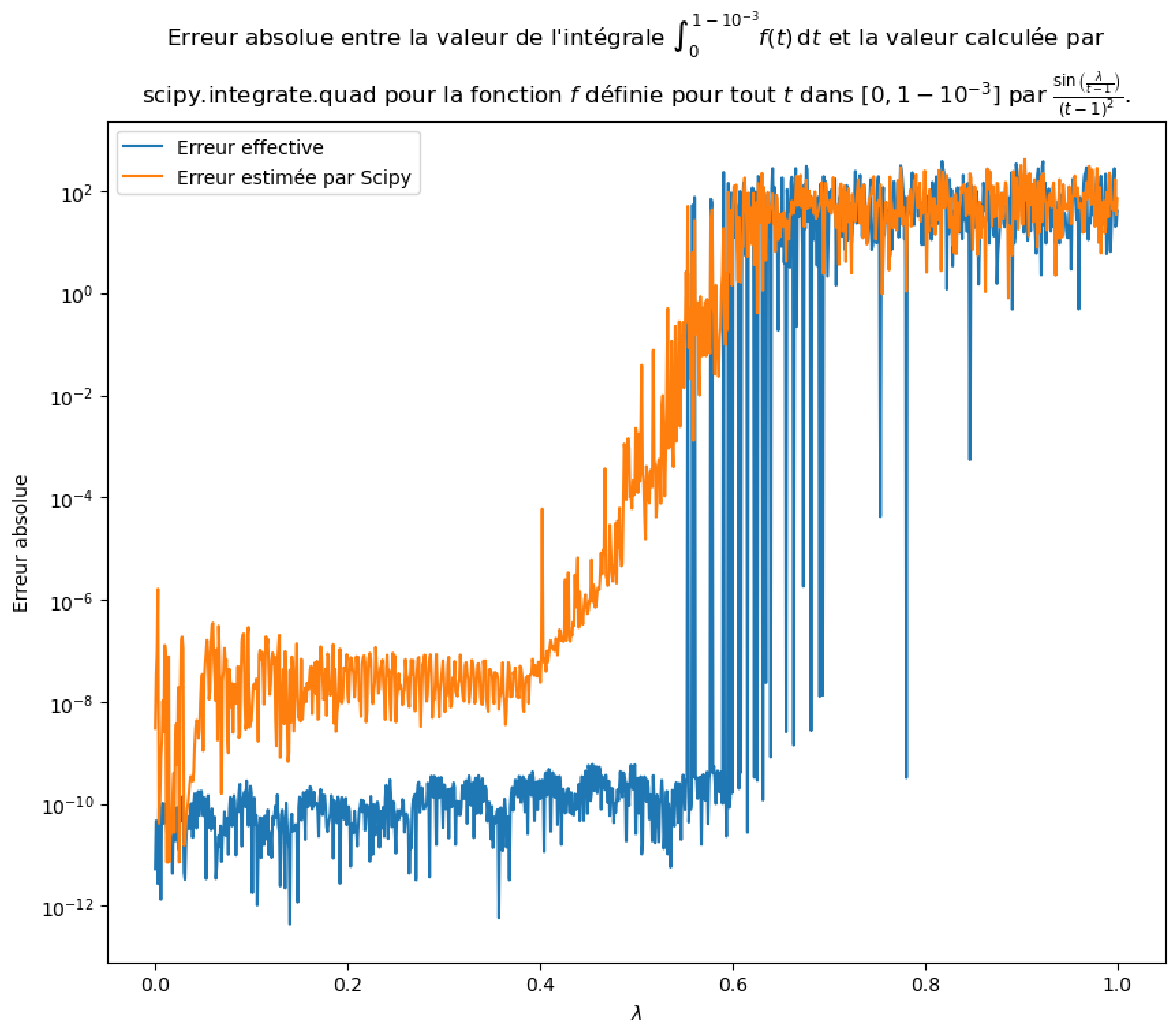
Commençons par un exemple qui nous permettra d'évaluer le comportement de cette méthode numérique face à de fortes variations. Soit f la fonction définie pour tout $t \in \mathbb{R} \setminus \{1\}$ par

$$f(t) = \frac{\sin\left(\frac{\lambda}{t-1}\right)}{(t-1)^2}$$

qui est facilement intégrable par changement de variable $u = \frac{\lambda}{t-1}$, ce qui nous donne une primitive de f,

$$F(t) = \frac{1}{\lambda}\cos\left(\frac{\lambda}{t-1}\right)$$

Bien sûr, étant donné que la valeur 1 est exclue à cause du domaine de notre fonction, nous prendrons l'intégrale jusqu'à une valeur légèrement en-dessous de 1 pour éviter une explosion de la valeur numérique.



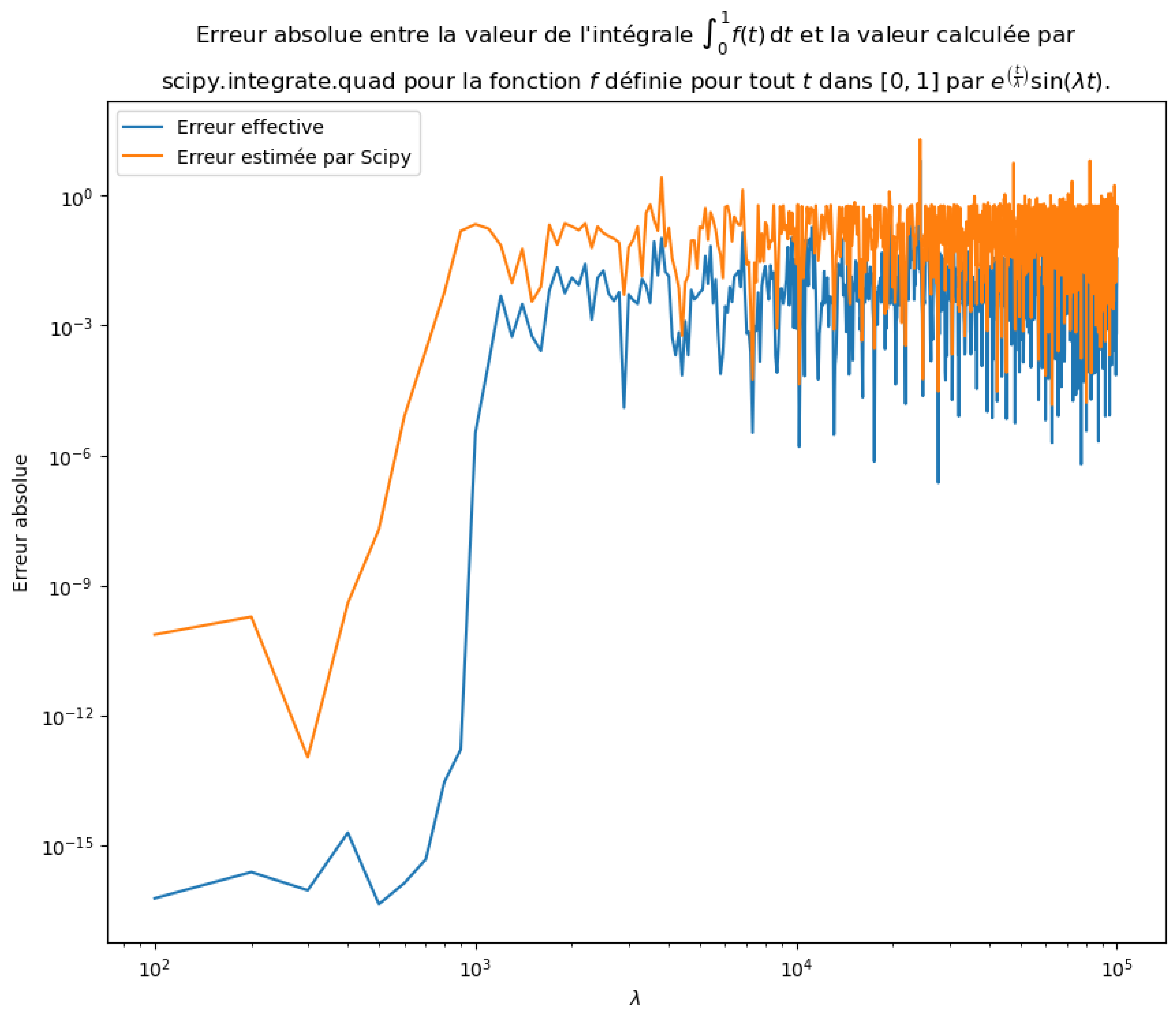
Sur ce graphique, nous pouvons voir que lorsque λ est proche de 0 et inférieur à environ 0.6, nous avons une très bonne estimation, les erreurs sont contenues entre 10^{-12} et 10^{-10}, cependant lorsque λ augmente, on constate une augmentation importante de l'erreur, qui varie de la même manière qu'elle soit réelle ou estimée par Scipy, une bonne nouvelle est que Scipy arrive toujours à donner une borne supérieure acceptable pour ces erreurs bien que cela ne soit pas toujours le cas, par contre, cela devient dangereux car nous arrivons à une erreur de 10^2 proche de λ entre 0.6 et 1. Nous ne pourrions donc pas nous fier à Scipy pour la précision des valeurs numériques pour cette fonction qui oscille énormément. On peut donc se méfier des estimations numériques pour des fonctions discontinues et non-intégrables de ce type, en plus d'être fortement oscillatoires.

Regardons une fonction moins sauvage, continue et dérivable mais oscillant toujours suffisamment. Soit f la fonction définie pour tout $t \in \mathbb{R}$ et tout $\lambda \in [10^2, 10^5]$ par

$$f(t) = e^{\left(\frac{t}{\lambda}\right)} \sin(\lambda t)$$

En utilisant surtout l'intégration par parties, nous trouvons qu'une primitive de f est définie par

$$F(t) = \frac{-\lambda e^{\frac{t}{\lambda}} \left(\lambda^2 \cos(\lambda t) - \sin(\lambda t)\right)}{\lambda^4 + 1}.$$



Nous avons les points nous assurant que la méthode d'intégration semble plus ou moins stable malgré le caractère oscillatoire de cette fonction, le tracé de l'erreur estimée par Scipy se trouve au-dessus de l'erreur effective, les deux suivant une variation quasi similaire, ce qui est rassurant lorsque nous souhaitons établir des intervalles de confiance. De plus, l'erreur est plus ou moins contrôlée, malgré une augmentation de l'erreur lorsque λ augmente, nous restons à priori en dessous de 10^0, ce qui est acceptable pour une telle fonction. Cela nous indique que l'intégration numérique est déjà plus adaptée pour des fonctions plus "sages".

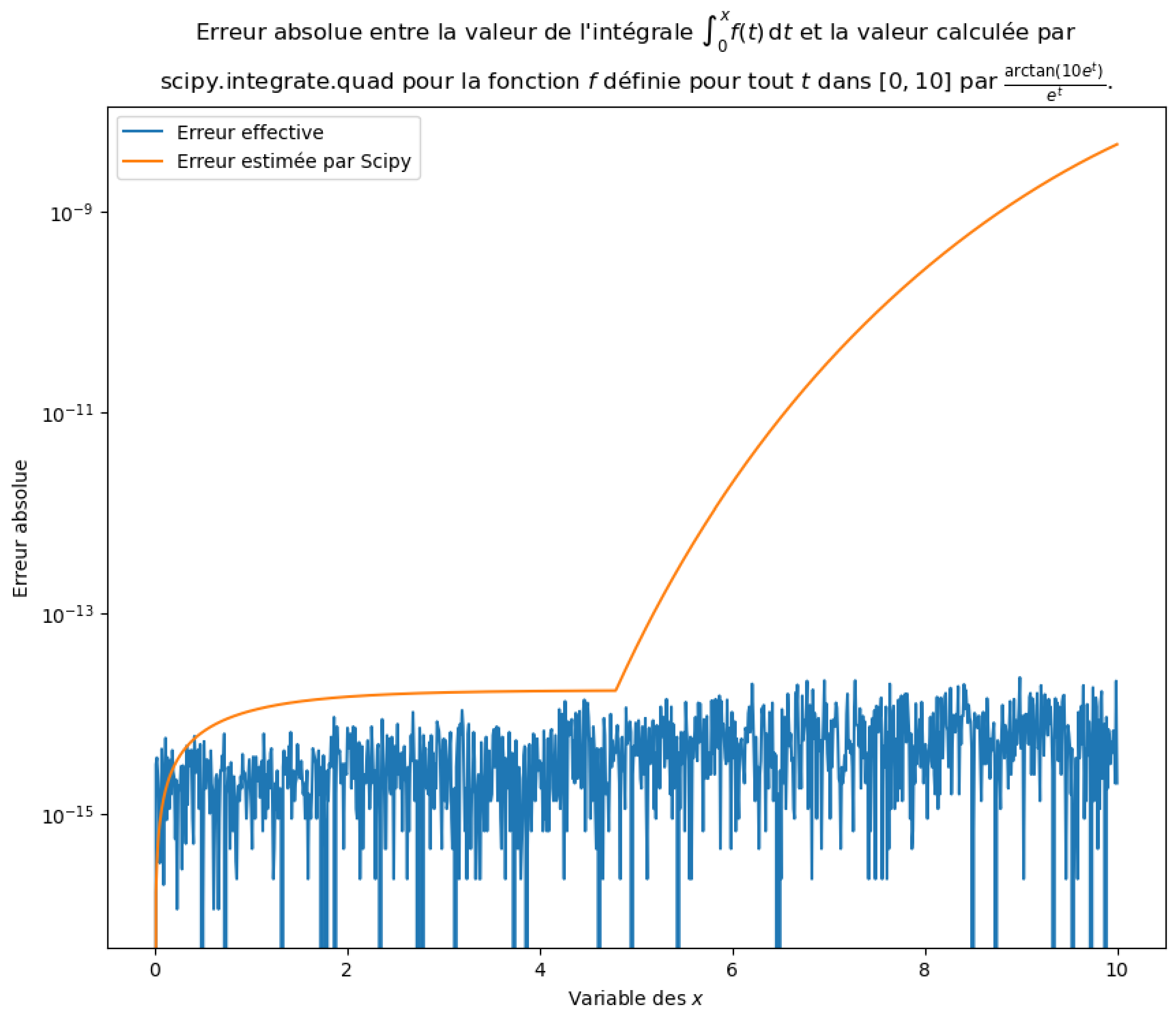
Finalement, au lieu de nous intéresser à la variation d'une valeur dans une fonction, regardons plutôt la variation de l'intégrale en fonction de la borne supérieure. Prenons également une fonction continue et dérivable, afin de vérifier si cela peut tout de même poser problème au niveau de l'intégration. Soit f la fonction définie pour tout $t \in [0, 10]$ par

$$f(t) = \frac{\arctan(10e^t)}{e^t}$$

Par le choix du paramètre, les données analysées seront moins prédictibles, vu cette fonction n'est pas très sensible à un coefficient près donc si on avait remplacé le facteur 10 par λ, nous n'aurions rien aperçu de spécial.

En vérifiant par Sympy, nous trouvons qu'une primitive de f est définie par

$$F(t) = 10t - 5\ln(100e^{2t} + 1) - e^{-t}\arctan(10e^t)$$



Nous voyons que les résultats d'erreurs que nous obtenons sont vraiment bons. Le tout est globalement sous 10^{-9} ce que nous pouvons considérer comme précis, et encore plus lorsque nous regardons l'erreur effective qui ne dépasse pas les 10^{-14}, environ. Pourtant, malgré le côté assez stable de cette fonction, nous remarquons que passé $x = 4$, l'erreur estimée par Scipy augmente fortement, pourtant dans cette fonction rien n'indique une raison évidente car elle est quasi constante à 0 pour les $t \in [5, 10]$ (visuellement). Cela doit sans doute s'expliquer par le fonctionnement de la méthode QUADPACK utilisée par Scipy.