TP : méthode d'Euler pour résoudre une équation différentielle

Nous avons vu en TD le principe de la méthode d'Euler pour résoudre une équation différentielle.
L'objet de ce TP est de résoudre de façon numérique l'équation différentielle

$$(E): \; y' = y^2 \text{ avec la condition } y(0) = \frac{1}{2}$$

On s'intéressera à l'allure de la solution sur l'intervalle $\left[0; \frac{3}{2}\right]$.

1 Mise en oeuvre de la méthode d'Euler

1. À la main, construire les trois premiers points qu'on obtient avec un pas $h = 0,5$.
2. Programmer une fonction python qui prend en paramètre un entier naturel non nul n et qui provoque l'affichage de la courbe obtenue avec la méthode d'Euler pour n points.

2 Résolution de (E)

— Analyse : supposons que (E) admette une solution, supposons de plus que cette solution ne s'annule jamais.

1. Diviser (E) par y^2.
2. Résoudre (E).
3. En déduire l'espression de la solution cherchée, ainsi que son domaine de définition.

— Synthèse : vérifier que la fonction trouvée est bien une solution de (E) qui ne s'annule jamais.

3 Qualité de l'approximation

1. Sur un même graphique, faire figurer la solution exacte trouvée ainsi que l'approximation fournie par la méthode d'Euler avec différentes valeurs de n. La méthode d'Euler vous semble-t-elle efficace ?
2. On se propose maintenant de mesurer l'erreur commise lorsqu'on travaille avec n points.
   Notons $x_1 < \cdots < x_n$ la subdivision régulière de $[0; \frac{3}{2}]$.
   Pour tout $i \in [\![1; n]\!]$, on approche $f(x_i)$ par y_i qui est fourni par la méthode d'Euler.
   On va appeler *erreur maximale* le maximum de la quantité $|f(x_i) - y_i|$.

   a) Programmer une fonction erreur(n) qui, étant donné un entier naturel $n > 1$, renvoie l'erreur maximale obtenue avec n points.

   b) Programmer une procédure (c'est-à-dire une fonction qui ne renvoit rien) qui provoque l'affichage de l'erreur maximale en fonction du nombre de points (on considérera n de 10 à 1000 avec un pas de 10).