

Fonction exponentielle - Algorithme par balayage

Énoncé

La plupart des lignes électriques font circuler du courant alternatif. Certaines font circuler du courant continu à très haute tension qui occasionne moins de pertes que le courant alternatif, notamment lorsque les lignes sont immergées, mais aussi lorsque les distances sont très importantes. En 2014, la plus longue liaison électrique à courant continu en service dans le monde parcourt le Brésil en reliant la centrale hydro-électrique de Porto Velho à Araraquara dans l'état de Sao Paulo. Elle mesure environ 2 300 km ; sa puissance électrique initiale est de 3150 MW et le courant est transporté sous une tension de 600 kV.

La puissance électrique au bout des 2 300 km de la ligne est de 2 940 MW.

1. a. Calculer le pourcentage des pertes de puissance électrique sur la ligne Porto Velho – Araraquara.

b. Calculer le taux d'évolution moyen de la puissance électrique aux 100 km sur cette ligne électrique.

2. Dans cette question, la puissance électrique (en méga-watts) restant dans une certaine ligne électrique à courant continu au bout de x centaines de kilomètres est donnée par la fonction P définie sur $[0 ; +\infty[$ par $P(x) = 3\,150 \times 0,997^x$.

a. Montrer que la fonction P est décroissante sur $[0 ; +\infty[$.

b. Compléter la fonction `lmax` du script Python LONGMAX ci-contre qui renvoie la longueur maximale (arrondie au kilomètre près) d'une ligne électrique, à l'aide d'un algorithme par balayage, pour laquelle la perte de puissance reste inférieure à 8%.

Retrouver ce résultat par résolution graphique à l'aide de la calculatrice puis par le calcul en utilisant le logarithme décimal.



Crédit photo : www.pexels.com – Ali Arapoğlu

```

ÉDITEUR : LONGMAX
LIGNE DU SCRIPT 0002
from math import *

def lmax():
  p=3150
  x=0
  while .....:
    x=x+.....
    p=3150*0.997**x
  return floor(x*100)-1
  
```

Fns... | a A # | Outils | Exéc | Script

NORMAL FLOTT AUTO RÉEL RAD MP

$$\left(\frac{2940}{3150}\right)^{\frac{1}{23}}$$

0.9970048045

Rep-1

-0.0029951955

1.a. Perte de puissance

La perte sur cette ligne est de $3\,150 - 2\,940 = 210$ pour une puissance initiale de 3 150. La perte est donc égale à $\frac{210}{3150} \approx 0,067 = \frac{6,7}{100} = 6,7\%$.

1.b. Taux moyen de perte aux 100 km

Soit t le pourcentage d'évolution de la puissance électrique aux 100 km. Puisque la ligne mesure 2 300 km, cela correspond à 23 centaines donc la valeur de t vérifie l'équation : $(1 + t)^{23} = \frac{2940}{3150}$ ce qui permet d'obtenir

$t = \left(\frac{2940}{3150}\right)^{\frac{1}{23}} - 1 \approx -0,002995$ soit une perte moyenne d'environ 0,3 % par centaine de kilomètres. On utilise les touches \wedge et $\frac{\square}{\square}$ pour accéder à ces fonctionnalités sur la calculatrice.



Fonction exponentielle - Algorithme par balayage

2.a. Décroissance exponentielle

D'après le cours, puisque $3\,150 > 0$, la fonction P a le même sens de variation que la fonction $x \mapsto 0,997^x$ qui est décroissante sur $[0; +\infty[$ car $0 < 0,997 < 1$.

2.b. Longueur maximale

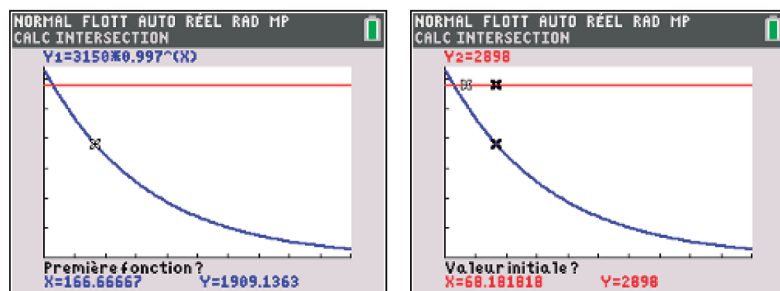
p représente la puissance, elle ne doit pas perdre plus de 8% ce qui se traduit par l'instruction `p>=3150*0.92`. La variable x correspond à des centaines de kilomètres, ainsi puisque l'on souhaite trouver une longueur maximale arrondie au km près, nous devons réaliser un balayage avec un pas de 0,01. Enfin en programmation, en raison du codage des nombres réels, nous sommes souvent confrontés à utiliser des arrondis, c'est le cas ici lors du renvoi de la valeur recherchée.

On exécute le script et on lance la fonction `lmax` en appuyant sur `var`.

```

PYTHON SHELL
>>> # Shell Reinitialized
>>> # L'exécution de LONGMAX
>>> from LONGMAX import *
>>> lmax()
2775
  
```

- Pour retrouver ce résultat à l'aide d'un graphique, nous entrons la fonction P dans Y_1 et la fonction constante égale à $3\,150 \times 0,92 = 2\,898$ dans Y_2 . On représente alors ces deux fonctions avec `graphe` après avoir configuré la fenêtre d'affichage. Puis on choisit le menu `calculs`, à l'aide des touches `2nde` `trace` et on sélectionne la commande 5: `intersection`. De retour au graphique, on valide le choix de Y_1 , celui de Y_2 et enfin la valeur initiale en se plaçant près du point d'intersection recherché :



- Enfin pour retrouver ce résultat par calcul, on résout l'inéquation $P(x) \geq 0,92 \times P(0)$ qui est équivalente à $0,997^x \geq 0,92$.

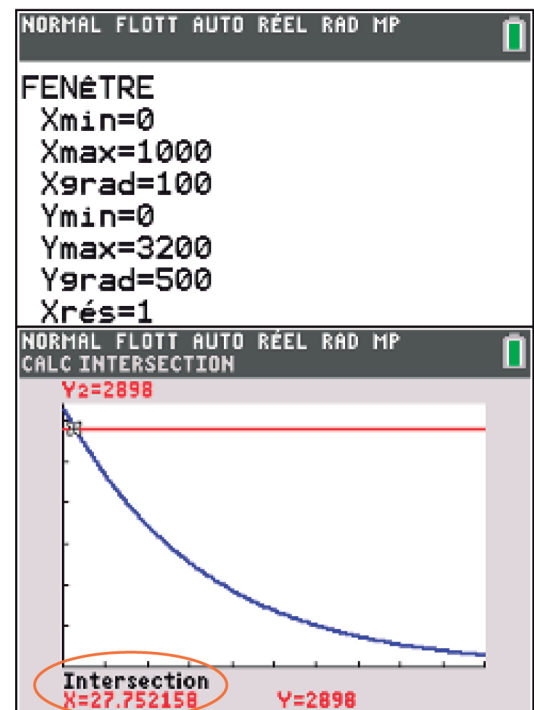
On utilise alors le logarithme décimal et puisque $\log(0,997) < 0$ car $0 < 0,997 < 1$ cette inéquation équivaut à $x \leq \frac{\log(0,92)}{\log(0,997)}$.

On trouve à nouveau une longueur maximale de la ligne électrique égale à 2 775 km arrondie au km près.

```

ÉDITEUR : LONGMAX
LIGNE DU SCRIPT 0010
from math import *

def lmax():
  p=3150
  x=0
  while p>=3150*0.92:
    x=x+0.01
    p=3150*0.997**x
  return floor(x*100)-1
  
```



```

NORMAL FLOTT AUTO RÉEL RAD MP
log(0.92)
log(0.997)
.....27.75215797
  
```