Travaux dirigés : Coûts

I. Définitions

I. A. Coût total

Le coût total C comprend l'ensemble des dépenses nécessaires à la production d'une certaine quantité x de produit. On le décompose généralement en :

- un coût fixe CF indépendant des quantités de production,
- et un coût variable CV qui est une fonction croissante du volume de production.
- Soit $C(x) = CF(x) + CV(x)$

I. B. Coût moyen

Le coût moyen de production, CM , mesure le coût par unité produite :
$CM(x) = \dfrac{C(x)}{x}$.

I. C. Coût marginal

Le coût marginal de production Cm mesure la variation du coût total pour une unité supplémentaire de production :
$Cm(x) = C(x+1) - C(x)$.

II. Exercice

Partie A :
Le tableau suivant donne les différents coûts journalier du produit.

C2 =0,0005*A2^3-0,05*A2^2+4*A2

	A	B	C	D	E	F
1	Quantités	Coûts fixes	Coût variable	Coût total	Coût moyen	Coût marginal
2	0		0,00 €			
3	1		3,95 €			

Le coût fixe journalier est de 98 euros.

1. À partir de la formule de la cellule C2 donner l'expression du coût variable suivant la quantité x produite.
2. Compléter le tableau en indiquant les formules utilisées en cellule
   - D2
   - E3
   - F2
3. Pour quelle quantité le coût moyen semble-t-il minimal ? Donner le coût marginal associé.

Partie B :
Avec Python le code suivant permet de mettre en évidence les différents coûts, le graphique illustre le coût total et la lecture du coût marginal. Saisir le code suivant et exécuter les commandes au fur et à mesure :

1. Compléter la fonction du coût marginal et la fonction du coût moyen :

```python
import matplotlib.pyplot as plt

#fonctions des différents coûts
def cout_fixe(x):
    return 98

def cout_variable(x):
    return round(0.0005*x**3-0.05*x**2+4*x,2)

def cout_total(x):
    return round(cout_fixe(x)+cout_variable(x),2)

def cout_marginal(x):
    return ..................

def cout_moyen(x):
    return ..................
```

couts.py

2. On donne l'algorithme et le programme pour obtenir la table du coût total (T est une liste de nombres) :

$T \leftarrow [\,]$

Pour i allant de 0 à n faire

ajouter l'élément $cout\_total(i)$ à T

fin pour

```python
#tables de valeurs
def table_cout_total(n):
    T=[]
    for i in range(0,n+1):
        T.append(cout_total(i))
    return T
```

couts.py

De la même manière faire la table du coût marginal.

3. On donne le code pour la représentation graphique du coût marginal à partir de la représentation du coût total :

```python
    x = []
    y = []
    for i in range(0, n+1):
        x.append(i)
        y.append(cout_total(i))
    plt.axis([0, n, 0,int(max(table_cout_total(n)))+1]) #défini la fenêtre graphique
    plt.scatter(x, y)
    for i in range(0,n):
        plt.plot([i+1,i+1],[cout_total(i),cout_total(i+1)])
        plt.plot([i,i+1],[cout_total(i),cout_total(i)])
    plt.show()
```

couts.py

Faire la représentation graphique pour 100 points puis 10 points : graph(100) puis graph(10).
Quelle ligne de code donne la représentation du coût marginal ?

Partie C :
On donne la fonction de coût par son expression avec $x \in [0\ ;\ 120]$, $C(x) = 0,0005x^3 - 0,05x^2 + 4x + 98$.
Sur GeoGebra,

- Construire la courbe de la fonction C
- Construire un curseur n variant de 0 à 120 avec un pas de 1 (nombre de pizzas)
- Construire les points A et B de la courbe d'abscisse n et $n+1$.
- Construire la tangente en A à la courbe de la fonction C.

1. Lire graphiquement le coefficient directeur de la tangente en A pour $n = 20$, $n = 50$ et $n = 100$.
   Que peut-on dire du coût marginal $Cm(n)$ de n produits et du nombre $C'(n)$?
   Pour la suite on admettra que $C'(n)$ est une bonne approximation du coût marginal $Cm(n)$.

2. Construire la courbe du coût moyen, donner le coût moyen minimal par lecture graphique.

3. Étude de fonction :
   (a) Exprimer $CM(x)$ $(x \neq 0)$ et $C'(x)$.
   (b) Montrer que $CM'(x) = \dfrac{(x-70)(x^2+20x+1400)}{1000x^2}$ et déterminer le tableau de signe de $CM'(x)$.
   (c) Dresser le tableau de variations du coût moyen CM.
   (d) On note x_0 la quantité pour laquelle le coût moyen est minimal. Montrer que $CM(x_0) = Cm(x_0)$.

correction du programme

```python
import matplotlib.pyplot as plt

#fonctions des différents coûts
def cout_fixe(x):
    return 98

def cout_variable(x):
    return round(0.0005*x**3-0.05*x**2+4*x,2)

def cout_total(x):
    return round(cout_fixe(x)+cout_variable(x),2)

def cout_marginal(x):
    return round(cout_total(x+1)-cout_total(x),2)

def cout_moyen(x):
    return round(cout_total(x)/x,2)

#tables de valeurs
def table_cout_total(n):
    T=[]
    for i in range(0,n+1):
        T.append(cout_total(i))
    return T

def table_cout_marginal(n):
    T=[]
    for i in range(0,n):
        T.append(cout_marginal(i))
    return T

#représentation graphique
def graph(n):
    x = []
    y = []
    for i in range(0, n+1):
        x.append(i)
        y.append(cout_total(i))
    plt.axis([0, n, 90,int(max(table_cout_total(n)))+1]) #défini la fenêtre graphique
    plt.scatter(x, y)
    for i in range(0,n):
        plt.plot([i+1,i+1],[cout_total(i),cout_total(i+1)]) #cout marginal=longueur segment
        plt.plot([i,i+1],[cout_total(i),cout_total(i)])
    plt.show()
```

couts_correction.py

Remarque : la commande exécuter la sélection du script permet de vérifier au fur et à mesure le travail du programme.