

Semaine n°12 du 15 décembre au 19 décembre

Informatique(Python) : cf exemples en annexe

- ⇒ Boucle `while` + Compteur.
- ⇒ Boucle `for` Calcul de somme.

Primitives et intégrales

- ⇒ Primitives usuelles (cf formulaire) et reconnaissance des composées.
- ⇒ Intégrale d'une fonction continue sur un segment : définition, expression à l'aide d'une intégrale de l'unique primitive d'une fonction s'annulant en un point.
- ⇒ Propriété de l'intégrale : linéarité, positivité (démonstration exigible), croissance (démonstration exigible).
- ⇒ Intégration par parties (démonstration exigible). Application au calcul d'une primitive du logarithme (démonstration exigible).
- Calcul d'intégrales du type $\int_a^b P(x) \cos(\alpha x) dx$, $\int_a^b P(x) \sin(\alpha x) dx$, $\int_a^b P(x) e^{\alpha x} dx$ où P est un polynôme.
- Calcul d'intégrales du type $\int_a^b \cos(\alpha x) e^{\beta x} dx$ ou $\int_a^b \sin(\alpha x) e^{\beta x} dx$ à l'aide de deux intégrations par parties.

Équations différentielles linéaires

- ⇒ Définitions : équations différentielles linéaires du premier ordre, équations différentielles linéaires du second ordre à coefficients constants et second membre non constant.
- ⇒ Théorème fondamental : Soit E une équation différentielle linéaire d'ensemble de solutions \mathcal{S} et E_h son équation homogène associée d'ensemble de solutions \mathcal{S}_h et f_0 une solution particulière de E , alors :

$$\mathcal{S} = \{g + f_0, g \in \mathcal{S}_h\}$$

- ⇒ Résolution des équations différentielles homogène linéaires du premier ordre.
- ⇒ Recherche d'une solution particulière des équations différentielles linéaires du premier ordre : solution évidente, méthode de la variation de la constante,
- ⇒ Exemples et résolution d'un problème de Cauchy du premier ordre.
- ⇒ pmp d'équations différentielles du 2nd ordre : solutions de l'équation homogène,

Remarques aux colleurs

- Merci aussi de poser une petite question d'informatique (cf Annexe).

Exemples de programmes informatiques

Exercice 1

Réaliser une fonction `DepasseValeur` prenant en paramètre un entier naturel M et renvoyant le plus petit entier naturel n tel que $2^n > M$.

```
def DepasseValeur(M):  
    n=0 #initialisation du compteur  
    while (2**n <=M):  
        n=n+1 #incrementation du compteur  
    return n
```

Exercice 2

Créer une fonction `somme` de paramètre n calculant $\sum_{k=0}^n k^3$

```
def somme(n):  
    S=0  
    for k in range(n):  
        S = S + k**3  
    return S
```

Exercice 3

Créer une fonction `suite` qui prend en entrée un entier naturel n et renvoie la valeur de u_n où (u_n) est la suite définie par :

$$\begin{cases} u_0 = \frac{1}{2} \\ \forall n \in \mathbb{N}, u_{n+1} = 3u_n + 2 \end{cases}$$

```
def suite(n):  
    u=1/2  
    for k in range(n):  
        u=3*u+2  
    return u
```